DEPARTMENT OF MECHANICAL ENGINEERING

School of Engineering and Architecture

UNIVERSITY OF MIAMI

GREAT SEAL UNIVERSITY OF MIAMI
CORAL GABLES FLORIDA 1926

Coral Gables, Florida 33124

CR-154624
May 1978

Volume III: Free Surface Models

Three Dimensional Thermal Pollution Models

National Aeronautics and
Space Administration

John F. Kennedy Space Center

Volume III: Free Surface Models

Three Dimensional Thermal Pollution Models

NASA Contractor Report CR-154624

Contract NAS10-8926
May 1978

Samuel S. Lee and Sabrata Sengupta

NASA CR-154624

KSC-TR 43-2

THREE-DIMENSIONAL THERMAL POLLUTION MODELS
VOLUME III - FREE SURFACE MODELS

By

Samuel S. Lee and Subrata Sengupta

Prepared for:
National Aeronautics and Space Administration
(NASA Contract NAS10-8926)

Department of Mechanical Engineering
School of Engineering and Environmental Design
University of Miami
Coral Gables, Florida

May, 1978

(i) PREFACE

This volume is the third of a three volume set presenting the description and program documentation of a mathematical model package for thermal pollution analyses and prediction. Two sets of programs, both in the free-surface formulation, are presented and clearly explained in this volume. These programs were developed by the Thermal Pollution Group at the University of Miami, and were funded by NASA, thus the program names NASUM II and NASUM III were given to reflect this joint effort.

These models are three-dimensional and time dependent using the primitive equation approach. They have sufficient generalality in programing procedure to allow application at sites with diverse topographical features. Both programs predict surface height variations; velocity field and temperature field for the "complete field". In the case of NASUM II a far-field formulation is used without including the plant thermal discharge; and in the case of NASUM III, a horizontal stretching is used to take account of the plant thermal discharge, and also to include far-field influences such as varying tide and ambient currents at points sufficiently far from the point of discharge.

These volumes are intended as user's manuals and, as such, present specific instructions regarding data preparation for program execution and specific simple problems.

TABLE OF CONTENTS, VOLUME III

(ii) LIST OF SYMBOLS

The following list of symbols which are obtained from Volume 1 are presented here for convenience.

A_1	First term in $\eta_o(t)$, which is defined below
A_2	Coefficient of second term in $\eta_o(t)$
C_o	phase velocity of surface gravity waves, $\sqrt{gH}$
C_p	specific heat at constant pressure
f	Coriolis parameter
g	acceleration due to gravity
h	depth relative to the mean water level
H	depth contour relative to free surface, $h + \eta$
I	grid index in x-direction or α direction
J	grid index in y-direction or β direction
K	grid index in z-direction or σ direction
k	thermal conductivity
K_H	horizontal eddy viscosity
K_V	vertical eddy viscosity
K_s	surface heat transfer coefficient
l	width of bay at ocean-bay interface
L	horizontal length scale
P	pressure
P_s	surface pressure

T	Temperature
T_{air}	Air temperature
T_{amb}	Water ambient temperature
T_e	Equilibrium temperature
t	time
t_ϕ	time lag in $\eta_o(t)$, which is defined below
u	velocity in x-direction (dimensional)
v	velocity in y-direction (dimensional)
v_o	Amplitude of inlet tidal velocity $v_o(t)$
w	Velocity in z-direction (dimensional)
x	Horizontal coordinate
y	Horizontal coordinate
z	Vertical coordinate

<u>Greek Letters</u>

α	horizontal coordinate in stretched system, $= x$
β	horizontal coordinate in stretched system, $= y$
σ	vertical coordinate in stretched system, $= z/H$
ρ	density
ϕ	phase angle in tidal current velocity
Ω	transformed vertical velocity
ω	angular frequency of tidal wave
τ_{xz}	surface shear stress in x-direction
τ_{xz}	surface shear stress in y-direction

η free surface elevation

$\eta_o(t)$ inlet tide level $= A_1 + A_2 \cos \omega(t+t_\phi)$

<u>Horizontal Stretching Parameters</u>

X horizontal stretching coordinate in x-direction

Y horizontal stretching coordinate in y-direction

X' $\frac{dX}{d\alpha}$

X'' $\frac{d^2Y}{d\alpha^2}$

Y' $\frac{dY}{d\beta}$

Y'' $\frac{d^2Y}{d\beta^2}$

a the distance at which minimum step size is desired in x-direction (see transformation relation below)

b the distance at which minimum step size is desired in y-direction (see transformation relation below)

a, b, c_1, c_2, c_3, c_4, d and e are related and defined by the following relationships

$$\alpha = a + c_1 \text{ Sinh}' \{c_2(X-d)\}$$

$$\beta = b + c_3 \text{ Sinh}' \{c_4(Y-e)\}$$

1. INTRODUCTION

This volume contains descriptions of the NASUM II and NASUM III computer programs together with instructions on how to operate these programs. As outlined in Volume I of this report, NASUM II is a three-dimensional, time dependent, free-surface model intended for use in large domains where rather coarse resolution is satisfactory. In NASUM II the horizontal distance between nodes in the rectangular grid employed in the model is the same throughout the domain. NASUM III has the same basic three-dimensional, free-surface character as NASUM II but includes a form of "horizontal stretching" which provides fine resolution in some parts of the domain and coarse resolution in other parts. The horizontal distance between nodes in the rectangular grid employed in the model is consequently a function of location in the domain.

The program descriptions, associated algorithms, flow charts, program symbols, choice of input data, and sample problems for NASUM II and NASUMIII are contained herein for the ready access of the computer programs by the user. Note, that the governing equations, approximations, simplifying assumptions, and numerical methods of solution are presented in Volume I.

NASUMII (the far-field version of the free surface model) has been applied to South Biscayne Bay, Florida and is presented with computer results in Lee and Sengupta's (1977) report on Three-Dimensional Thermal Pollution Models and in a paper by Sengupta, Lee and Miller (1977). The South Biscayne Bay is a relatively shallow estuary with the principal driving mechanism being tidal flux at the ocean-bay interface, although wind effects are clearly evidenced in the northern part of the bay.

NASUM III (the horizontally stretched version of the free surface model) has been applied to Hutchinson Island, St. Lucie, Florida, which is a coastal site with a submerged discharge. Horiztonal stretching was used in order to obtain resolution in the neighborhood of the discharge, while at the same time a large horizontal domain could be covered. If a constant grid size had been used, this would have required an excessively large number of grid points to cover the same extent of the boundaries of the domain. Therefore, in order to circumvent this problem, a hyperbolic sine (SINH) stretching transformation was used in both the horizontally lateral and transverse directions, respectively, to obtain a small grid size in the neighborhood of the discharge and an increasingly larger grid size as distance from the discharge point increased. Waldrop and Farmer (1973) suggested a tangent (TAN) stretching transformation; however, the SINH stretching transformation was found to have advantages in this study. The details of the comparison between the tangent and sinh stretching formulas are presented in Volume I. Lee and Sengupta (1977) and Tsai (1977) present the results of this Hutchinson Island investigation.

The effects of variable bottom topography, spatio-temporal free surface variations, surface heat transfer based on the equilibrium temperature concept introduced by Edinger and Geyer (1971), tide level variation, resultant ambient currents, and meteorological conditions have been factored into these models. In addition, turbulence has been modelled by using the eddy transport concept, and the effects of baroclinicity have been included. Again, the user should refer to Volume I for the complete mathematical formulations, approximations and assumptions, and the numerical methods of solution.

2. Program Descriptions

This section presents the computer program algorithms and associated flow charts (in standard notation. c.f. Murrill and Smith (1975) for the NASUM II and NASUM III.

2.1 NASUM II (The Far Field Model)

2.1.1 Description of Program Algorithm

The program algorithm for NASUM II is a follows:

a) Integrate the surface height equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS).

b) Integrate the u-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS) with DuFort-Frankel differencing applied to the vertical momentum diffusion term as given, for example, in Roache (1972).

c) Integrate the v-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS) with DuFort-Frankel differencing applied to the vertical momentum diffusion term.

d) The equivalent vertical velocity, Ω, is then computed by knowing H, u and v. The spatial integration is performed by applying Simpson's rule. (c.f. Crandell (1955)).

e) The energy equation is then intregrated over time using forward-time, central-space (FTCS) throughout.

f) The density is calculated from the equation of state.

g) The pressure field is calculated from the hydrostatic equation using the trapezoidal rule for spatial integration.

h) Then, the value of computed real time (or simulation time) is checked and the steps a) through g) repeated if so desired. Reference to the flow chart presented in Fig. 1 will clarify this last step in the program algorithm.

2.1.2 Flow Chart

The flow chart for NASUM II is presented in Fig. 1.

2.2 NASUM III (The horizontally stretched model)

2.2.1 Description of Program Algorithm

The program algorithm for NASUM III is as follows:

a) Integrate the surface height equation using forward-time, central-space differencing initially (FTCS), and , thereafter, central-time, central-space is used (CTCS).

b) Integrate the u-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central -space is used (CTCS) without DuFort - Frankel differencing applied to the vertical momentum diffusion then, since the vertical diffusion term does not govern the time step value as it does for a shallow estuary.

c) Integrate the v-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS).

d) The equivalent vertical velocity, Ω, is then computed by knowing H, u and v. However, for a submerged discharge Ω at the bottom of the basin is not zero (c.f. Volume I). The spatial integration is performed by applying the trapezoidal rule.

e) The energy equation is then integrated over time using forward-time, central-space (FTCS) initially, and, thereafter, central-time, central-space (CTCS) is used.

f) The density is calculated from the equation of state.

g) The pressure field is calculated from the hydrostatic equation using the trapezoidal rule for spatial integration.

h) Then, the value of computed real time (or simulation time) is checked and the steps a) through g) repeated if so desired. Reference to the flow chart presented in Fig. 2 will clarify this last step in the program algorithm.

2.2.2 Flow Chart

The flow chart for NASUM III is presented in Fig. 2.

3. LIST OF PROGRAM SYMBOLS

3.1 Symbols Common to Far-Field and Horizontally Stretched Model Programs

This section presents in alphabetical order the program symbols, in FORTRAN language, and their definition for those symbols which are common to the Far-Field and Horizontally Stretched Models. In many cases the definition is shortened by referring to algebraic symbols already defined in section (ii) of this volume.

A

A1 : first term in $\eta_o(t)$

A2 : Coefficient of second term in $\eta_o(t)$

B

BH : B_H

BV : B_V

C

CI : Coefficient of inertia term in momentum equations

CC : Coefficient of Coriolis term in momentum equations

CP : Coefficient of pressure term in momentum equations

CH : Coefficient of horizontal diffusion term in momentum equations

CV : Coefficient of vertical diffusion term in momentum equations

D

D(I,J,K)	:	$u(\alpha,\beta,\sigma)$ at $t = t + \Delta t$
DT	:	time step, Δt
DX	:	Grid size in α - direction, $\Delta\alpha$
DY	:	Grid size in β - direction, $\Delta\beta$
DZ	:	Grid size in α - direction, $\Delta\sigma$
DUM	:	$\frac{\sigma}{H} \int_0^1{}' \{ \frac{\partial(Hu)}{\partial\alpha} + \frac{\partial(Hv)}{\partial\beta} \} d\sigma$
DIHUTX	:	$\frac{\partial(HuT)}{\partial\alpha}$
DIHUUX	:	$\frac{\partial(Huu)}{\partial\alpha}$
DIHUVX	:	$\frac{\partial(Huv)}{\partial\alpha}$
DIHUVY	:	$\frac{\partial(Huv)}{\partial\beta}$
DIHUX	:	$\frac{\partial(Hu)}{\partial\alpha}$ at $k = k$
DIHUX1	:	$\frac{\partial(Hu)}{\partial\alpha}$ at $k = k-1$
DIHVTY	:	$\frac{\partial(HvT)}{\partial\beta}$
DIHVVY	:	$\frac{\partial(Hvv)}{\partial\beta}$
DIHVY	:	$\frac{\partial(Hv)}{\partial\beta}$ at $k = k$
DIHVY1	:	$\frac{\partial(Hv)}{\partial\beta}$ at $k = k-1$
DIPX	:	$\frac{\partial p}{\partial\alpha}$
DIPY	:	$\frac{\partial p}{\partial\beta}$

D1TX	:	$\frac{\partial T}{\partial \alpha}$
D2TX	:	$\frac{\partial^2 T}{\partial \alpha^2}$
D1TY	:	$\frac{\partial T}{\partial \beta}$
D2TY	:	$\frac{\partial^2 T}{\partial \beta^2}$
D2TZ	:	$\frac{\partial^2 T}{\partial \sigma^2}$
D1UX	:	$\frac{\partial u}{\partial \alpha}$
D2UX	:	$\frac{\partial^2 u}{\partial \alpha^2}$
D1UY	:	$\frac{\partial u}{\partial \beta}$
D2UY	:	$\frac{\partial^2 u}{\partial \beta^2}$
D1UWZ	:	$\frac{\partial (u\Omega)}{\partial \sigma}$
D2UZ	:	$\frac{\partial^2 u}{\partial \sigma^2}$
D1VX	:	$\frac{\partial v}{\partial \alpha}$

	D2VX	: $\frac{\partial^2 v}{\partial \alpha^2}$
	D1VY	: $\frac{\partial v}{\partial \beta}$
	D2VY	: $\frac{\partial^2 v}{\partial \beta^2}$
	D1VWZ	: $\frac{\partial (v\Omega)}{\partial \sigma}$
	D2VZ	: $\frac{\partial^2 v}{\partial \sigma^2}$
	D1WTZ	: $\frac{\partial (\Omega T)}{\partial \sigma}$
<u>E</u>	E(I,J,F)	: $v(\alpha,\beta,\sigma)$ et $t=t+\Delta t$
	ETA(I,J)	: $\eta(\alpha,\beta)$
	ETAX(I,J)	: $\frac{\partial \eta}{\partial \alpha}$
	ETAY(I,J)	: $\frac{\partial \eta}{\partial \beta}$
<u>F</u>	FF	: f
<u>G</u>	G(I,J,K)	: $u(\alpha,\beta,\sigma)$ at $t=t$
	GR	: g
<u>H</u>	H(I,J,K)	: $u(\alpha,\beta)$ at $t=t$
	H(I,J)	: $H(\alpha,\beta)$ at $t=t$ in energy equation
	HDUM(I,J)	: $-\int_0^1 \{ \frac{\partial (Hu)}{\partial \alpha} + \frac{\partial (Hv)\}}{\partial \beta} \alpha\sigma$

HI(I,J): $h(\alpha,\beta)$

HN1(I,J): $H(\alpha,\beta)$ at $t=t+\Delta t$ in energy equation

HS: Ks

HT(I,J): $H(\alpha,\beta)$ at $t=t-\Delta t$ in momentum equations

HTD(I,J): $H(\alpha,\beta)$ at $t=t$ in momentum equations

HTE(I,J): $H(\alpha,\beta)$ at $t=t+\Delta t$ in momentum equations

HTMIN: Minimum value of $H(\alpha,\beta)$ in bay

HX(I,J): $\frac{\partial h}{\partial \alpha}$

HY(I,J): $\frac{\partial h}{\partial \beta}$

<u>I</u>

I: Index in α - direction

I1: Lower index for inlet along MAR=1*

I2: Upper index for inlet along MAR=1

I3: Lower index for outlet along MAR=2

I4: Upper index for outlet along MAR=2

I5: Index for inlet along MAR=3

I6: Index for outlet along MAR=4

IBAY: Parameter which when specified either provides a <u>constant</u> time step for a shallow bay or for a deep bay.
(=o for shallow bay; =1 for deep bay)

IHITE: =o for specifying initial surface
=1 for <u>not</u> specifying initial surface

IN: Number of grid points in α - direction

INLET: =1 for inlet along MAR=1
=2 for inlet along MAR=3

IRUN: =o for first run
=1 thereafter

*NOTE: The MAR(I,J) matrix is explained in section 6.1.3. Fig.3 illustrates the location of I1 through I6 and J1 through J6.

J J: Index in β - direction

J1: Index for inlet along MAR=1

J2: Index for outlet along MAR=2

J3: Lower index for inlet along MAR=3

J4: Upper index for inlet along MAR=3

J5: Lower index for outlet along MAR=4

J6: Upper index for outlet along MAR=4

JN: Number of grid points in β - direction

K K: Index in σ- direction

KH: K_H

KN: Number of grid points in σ - direction

KV: K_V

L L: Index of time cycle without energy equation

LL: Index of time cycle with energy equation

LN: Number of time cycles without energy equation

LNI: Number of time cycles with energy equation

M M: Parameter for either specifying V_o (t) at the inlet or specifying η_o (t) at the inlet (=1 for V_o,=2 for η_o case)

MAR(I,J): Numbering system for grid system - used to distinguish between different boundary finite difference schemes.

P P(I,J,K): $P(\alpha,\beta,\sigma)$

R RO(I,J,K): $\rho(\alpha,\beta,\sigma)$ for variable density case

RR: $\rho(\alpha,\beta,\sigma)$ for constant density case

T T(I,J,K): $T(\alpha,\beta,\sigma)$ at t=t

TA: Te

TAUX: τzx

TAUY: τzy

TFLAT: Time interval from initial flat surface ($\eta_o = o$) to some desired hour

THT: Time interval from start-up to high tide

TI: Initial temperature for isothermal bay

TNI(I,J,K): $T(\alpha,\beta,\sigma)$ at $t = t + \Delta t$

TPH: Time lag for V_o (t)

TPHI: Time lag for η_o (t)

TT: TTOT + TTOTI

TTOT: Total run time without energy equation

TTOTI: Total run time with energy equation

U U(I,J,K): $u(\alpha,\beta,\sigma)$ at $t = t - \Delta t$

V V(I,J,K): $v(\alpha,\beta,\sigma)$ at $t = t - \Delta t$

VO: V_o amplitude of $V_o(t)$ at inlet

W W(I,J,K): $\Omega(\alpha,\beta,\sigma)$

WUD: $-\frac{1}{H}\int_o^\sigma \left[\frac{\partial(Hu)}{\partial\alpha} + \frac{\partial(Hv)}{\partial\beta}\right] d\sigma$

WZ(I,J,K): $\omega(\alpha,\beta,\sigma)$

3.2 Additional Symbols for Horizontally Stretched Model Program

This section presents in alphabetical order the program symbols, in FORTRAN language, and their definition for those additional symbols for the Horizontally Stretched Model. Again, symbols not defined here have already been defined in section (ii).

A

A: Value of $(X - d)/C_1$

B

B: Value of $(Y - e)/C_3$

D

DEEX: Value of C_1

DEEY: Value of C_3

DELX; Grid size in α- direction, $\Delta\alpha$

DELY: Grid size in β - direction, $\Delta\beta$

DHDX: $\frac{\partial H}{\partial \alpha}$ at t=t

DHDY: $\frac{\partial H}{\partial \beta}$ at t=t

E

EEEX: Value of d

EEEY: Value of e

H

HK: K_s

HTX: $\frac{\partial H}{\partial \alpha}$

HTY: $\frac{\partial H}{\partial \beta}$

T

T(I,J,K): $T(\alpha,\beta,\sigma)$ at $t=t - \Delta t$

TN(I,J,K): $T(\alpha,\beta,\sigma)$ at t=t

TF(I,J,K): $T(\alpha,\beta,\sigma)$ at $t=t + \Delta t$

TAIR: Air temperature

TAM: Ambient temperature of water body

U

UM: $u(\alpha,\beta,\sigma)$

V

VM: $v(\alpha,\beta,\sigma)$

W

WH: $w(\alpha,\beta,\sigma)$

X

XX· $\frac{dX}{d\alpha}$

XXX: $\frac{d^2X}{\alpha\alpha^2}$

$\underline{Y}$

YY: $\frac{d\,Y}{\alpha\beta}$

YYY: $\frac{d^2\,Y}{\alpha\beta^2}$

4. MAIN PROGRAMS

This section presents a detailed description of the main programs for the NASUM II and NASUM III. The main programs themselves appear in Section 7.1.

4.1 NASUM II (Far-Field Main Program)

The following main program outline and associated description is for the far-field version of the free surface model. The main program name is FMAIN, and appears in Section 7.1.

a) Specify number of grid points, IN, JN and KN in PARAMETER statement (although the geometry of the domain of solution under consideration will not cover all the grid points; MAR(I,J)=0 covers range of grid points outside the domain of solution, where MAR(I,J) is constructed as shown in Fig. 5 for application to the South Biscayne Bay).

b) Specify IRUN=0 or 1. The value 0 is used for the first run only, and 1 is used thereafter.

For IRUN = 0, READ2 and INITIA are used.

For IRUN = 1, READ1 is used.

c) Specify LN, LN1, M, INLET, IBAY, IHITE, I1, I2, I3, I4, I5, I6, J1, J2, J3, J4, J5, J6, VO, TPH, TPH1, A1, A2, HTMIN, THT, TFLAT, CI, CC, CP, CH. CV, GR, FF, RR, DX, DY, DZ, KH, KV, BH, BV, TI. See section 3.1 for definition of these symbols, and refer to section 6.1.4, to follow, for a sample input of these parameters.

d) Specify TAUX, TAUY, TA, and HS, as defined in section 3.1, <u>each hour</u>.

e) Specify DT as defined in section 3.1. (in seconds)

f) For L=1, TTOT=0.0: Energy equation is <u>not</u> coupled to the system of governing equations. The following subroutines are used:

HEIGHT
TIDE or VEL
DATA
UVVEL
WVEL
PRES
ETT

g) For L>1: Energy equation is <u>not</u> coupled to the system of governing equations, but central-time is used now after the first time step has been executed (for L=1).

HEILN
TIDE or VEL
DATA
UVVELN
WVEL
PRES
ETT
OLDHT
OLDUV

h) For LL>1, TT=TTOT + TTOT1 > 0: Energy equation is coupled to the system of governing equations. The following subroutines are used:

HEILN
TIDE or VEL
TIDAL
DATA
UVVELN
WVEL
PRES
ETT
OLDHT
OLDUV
TEMP
OLDT

i) After the final time cycle is computed, the following subroutines are used for printing and storing on magnetic tape:

PRPARA
PRETA
PRUV
WW
PRW
PRTEMP
STORE

4.2 NASUM III (Horizontally Stretched Main Program)

The following main program outline and associated description is for the horizontally stretched version of the free surface model. The main program name is FMAIN, and appears in Section 7.1.

a) Specify number of grid points, JN, JN and KN in PARAMETER statement for the domain of interest.

b) Read in all the data required and logic parameters IRUN, LN, CI, CC, CP, CH, CV, GR, FF, RR, HK, DX, DY, DZ, KH, KV, BH, BV, TAUX, TAVY, TAIR, DT, DELX, DELY, DEEX, DEEY, EEEX, EEEY. See sections 3.1 and 3.2 for definition of these symbols; and refer to Section 6.2.4, to follow, for a sample input of these parameters.

c) Generate a two-dimensional matrix MAR(I,J) for locating the position of the points in the domain.

d) Initialize all the necessary quantities, as defined in Section 3.1; specify the discharge conditions and bottom topography.

e) Convert the real vertical velocities W into the transformed sigma coordinate vertical velocities, Ω.

f) Calculate the horizontal stretching parameters, X', X'', Y', Y'' for the set of governing equations.

g) Calculate the new predicted dependent variables.

h) Store the data and new predicted dependent variables on magnetic tape and print out these values at the desired time step.

i) For L=1, TTOT=0.0: Forward-time differencing is used. The following subroutines are used:

HEIGHT
UVVEL
TEMP
PRES
ETT

j) For L>1 or TTOT>DT: Central-time differencing is used. The following subroutines are used:

HEILN
UVVELN
WVEL
TEMPN
PRES
ETT
OLDUVT
OLDUV

k) After the final time cycle is computed, the following subroutines are used for printing and storing on magnetic tape:

STORE
PRPARA
PRETA
PRUV
PRW
PRTEM

5. INPUT DATA

The data that is required for the execution of the main program in either NASUM II or NASUM III is called Input Data. The data required is listed in the order it appears in the respective programs, and the corresponding FORMAT (in FORTRAN language) is given corresponding to each data symbol. Section 5.1.1 lists the data input required for running NASUM II, and Section 5.2.1 lists the data input required for running NASUM III. The actual calculation required for several of the input data is given in Section 6.1.3 for NASUM II, and in 6.2.3 for NASUM III. Note, the data input symbols have already been defined in Section 3 of this volume.

5.1 NASUM II(Far-Field Model)

The following number of computer data cards, with proper FORMAT, is now given in order as they appear in the main program in order to execute NASUM II (refer to Section 3.1 for definition of these FORTRAN symbols). The data that must be calculated beforehand is given in Section 6.1.3.

5.1.1 DATA REQUIRED AND FORMAT

CARD NO.	DATA	FORMAT
1	IRUN	I5
2	LN	I5
3	LN1	I5
4	M	I5
5	INLET	I5
6	IBAY	I5
7	IHITE	I5
8	I1,I2,I3,I4,I5,I6	6I5
9	J1,J2,J3,J4,J5,J6	6I5
10	VO,TPH,TPH1,A1,A2	Free
11	HTMIN,THT	Free
12	CI,CC,CP,CH,CV	Free
13	GR,FF,RR	Free
14	DX,DY,DZ	Free
15	KH,KV	Free
16	BH,BV	Free
17	TI	Free
18	DTAUX(13)**	Free
19	DTAUY(13)	Free
20	DTA(13)	Free
21	DHS(13)	Free
22	DT	Free

**NOTE: 13 values of TAUX and TAUY and 13 values of TA and HS are read in for variation each hour. Note, the letter "D" preceeds previously defined symbols (Section 3.1), since this was necessary for computer convenience.

5.2 NASUM III (Horizontally Stretched Model)

The following number of computer data cards, with proper FORMAT, is now given in order as they appear in the main program in order to execute NASUM III(refer to Section 3.1 and 3.2) for definition of these FORTRAN symbols). The data that must be calculated beforehand is given in Section 6.2.3.

5.2.1 DATA REQUIRED AND FORMAT

CARD NO.	DATA	FORMAT
1	IRUN	I5
2	LN	I5
3	CI,CC,CP,CH,CV	Free
4	GR,FF,RR,HK	Free
5	DX,DY,DZ	Free
6	KH,KV,BH,BV	Free
7	TAUX,TAUY	Free
8	TAIR	Free
9	DT	Free
10	DELX	Free
11	DELY	Free
12	DEEX	Free
13	DEEY	Free
14	EEEX	Free
15	EEEY	Free

6. SAMPLE CASES

The following sample cases will illustrate and clarify to the user the proper choice of programs, subprograms (or subroutines), calculation of input parameters, sample input and sample output for NASUM II and NASUM III.

6.1 NASUM II (Far Field Model)

6.1.1 Problem Statement - Application to Biscayne Bay

Given Biscayne Bay, in Dade County, Florida, as an example application site, compute the surface heights, η, velocity field u, v, w, and the temperature distribution T at 2:00 P.M. knowing the meteorological data, the IR data base, and the tide data base for April 15, 1975. The IR data base is assumed synoptic at 2:00 P.M., the wind velocity and ambient temperature is known every hour, and the tide height, with respect to the mean water level, is known as a function of time at the ocean bay interface.

Use the NASUM II far-field, free surface model program to obtain the desired results

6.1.2 Choice of Subroutine Programs

Section 4.1 is followed in order to choose the proper subroutine programs for this sample case. Sections 6.1.3 and 6.1.4 to follow next, will clearly illustrate what steps the user must follow in order to obtain the desired results for this sample case.

6.1.3 Calculation of Input Parameters

a) Construct a three-dimensional grid system for the Biscayne Bay. Fig. 4 illustrates the horizontal grid for the bay superimposed on the actual geometry of the domain of interest. The governing equations have been transformed into the α,β,σ coordinate system, which maps the variable depth basin into a constant depth basin. Then , depending on the desired resolution of vertical structure, the number of vertical grid points is selected, that is KN. The values of IN and JN are then selected with consideration of desired horizontal resolution versus computer storage and overall computation time. Then, the next step is to specify IN, JN, KN in the main program, as mentioned in section 4.1. For this application IN=34, JN=11, KN=5.

b) Next, LN and LN1 as defined in section 3.1 are specified. First, however, the time step DT is computed based on the criteria given in Volume I. For the Biscayne Bay, DT is computed by the vertical momentum diffusion criterion:

$$DT = \Delta t < \frac{(z \text{ minimum})^2}{2} / K_v = \frac{(.25)^2(60.96\text{cm})^2}{2} / 5\text{cm}^2/\text{sec}.$$

Thus, DT = 10 sec. is chosen for this sample case. in order to ensure numerical stability.

For starting the program at η_o $(t=o) = 0$ for April 15, 1975 LN= 1342, LN1=1380, since the program is started at 6:26 a.m. and run without the energy equation to 10:10 am, at which time the IR data base is read in, as an initial condition, from subroutine TIDAL. Then, from 10:10 am to 2:00 P.M. the energy equation is included. This procedure ignores the effect of density currents on the momentum and surface height equations from 6:26 am to 10:10am. However, these density currents are quite small for the Biscayne Bay which is dominated by the wind and the tidal flux at the ocean-bay interface.

c) The eddy viscosity coefficients have been estimated by applying the "4/3 scaling law" to previously known water basin values used by other researchers.

d) Next, the matrix MAR(I,J) is constructed based on this particular grid system as shown in Fig. 5.

e) The depth matrix HI (I,J) is then constructed by specifying the depth below the mean water level, $h(\alpha,\beta)$ at each horizontal grid point.

f) The initial temperature matrix T (I,J,K) is constructed for the bay by first plotting the IR data base surface isotherms on the horizontal grid, and then interpolating to specify the temperature at each grid point (I,J). (See Fig. 6) The bay is shallow and well mixed vertically, hence, the vertical temperature variation is initially set equal to zero. Subroutine TIDAL reads in from data cards T(I,J,K).

g) The tidal current velocity ampli ude Vo is computed from the followin formula, assuming a 90° phase shift between the tide height and tidal current velocity at the ocean-bay interface, (Ippen 1966):

$$V_o = \left(\frac{2aCo}{h}\right)\left(\frac{2\pi l}{\lambda}\right)$$

where $a = |\eta_o|$ $= 37$ cm for April 15, 1975

$C_o = \sqrt{gh}$ $\cong 4.3 \times 10^2$ cm/sec

$l = 10\Delta\beta$ $= 16 \times 10^5$cm

$\lambda = C_oT$ $= 12C_o = 1.86 \times 10^7$ cm>> 1

<h> 190 cm, average at ocean-bay interface.

Thus, Vo= 90 cm/sec

h) TPH =3138.7 sec. is computed by letting Vo (t) =0 at 10:10 A.M.(high tide where $Vo(t) = Vo \cos\left[(t + TPH)\frac{2\pi}{T}\right]$ and t=o at 8:00 A.M. on 4/15/75.

TPH1 = 7800 + FLAT = 13440 sec.(where 7800 sec = 8:00 A.M. to 10:00 A.M.)

where TFLAT = 5640 sec., the time from $\eta_0(t) = 0$ at 6:26 A.M. for April 15, 1975 to the beginning of the $V_{\dot{0}}(t)$ run at 8:00 AM on April 15, 1975.

i) The equilibrium temperature, TA, and the equilibrium coefficient of surface heat transfer HS, are computed following Harleman and Stolzenbach (1973). Appendix C presents these formula.

j) The wind stresses TAUX, TAUY are determined as shown in Appendix A.

6.1.4 Sample Input

The far-field solution is obtained by either specifying Vo(t) or $\eta_0(t)$ at the ocean-bay interface. These two sample cases will now be given:

6.1.4.1 Velocity Case

IRUN = 0

LN = 360 (begin at 0800 EST, 4/15/75, and compute until 0900 EST)

LN1=1

M=1 (Vo(t) specified at inlet)

INLET = 1 (ocean-bay interface along MAR=1)

IBAY = 0 (shallow bay), i.e., constant time step of 10sec is used even in shallow regions - this was done to avoid inordinately small time steps during low water.

IHITE = 1 (regression surface not read in at time of high tide)

I1, I2, I3, I4, I5, I6= 7, 16, 31, 33, 35, 35(refer to Fig. 3)

J1, J2, J3, J4, J5, J6= 11, 1, 12, 12, 12, 12(refer to Fig. 3)

Vo, TPH, TPH1, A1, A2 =-90, 3138.7, -13420, 11.8872, 37.1856

HTMIN, THT = 60.96, 7800.0

TFLAT = 0.0

CI, CC, CP, CH, CV=1., 1., 1., 1.,1.

GR, FF, RR = 980., .00006, 1.

DX, DY, DZ = 160,000., 160,000., .25

KH, KV = 10,000., 5.

BH, BV = 10,000., 5.

TI = 24.5

DTAUX (1) = -.37

DTAUY (1) = .15

DTA (1) = 31.7

DHS (1) = .00129

.

.

.

DT = 10

Next, IRUN = 1, LN = 420, LN1 = 1 (10:10am)

Next, IRUN = 1, LN = 1, LN1 = 1380 (2:00pm)

6.1.4.2 Tide Height Case

IRUN = 0

LN = 264 (begin at 0626 est, 4/15/75 and compute unitl 0710 est)

LN1= 1

M = 2 (η_o(t) specified at inlet)

INLET = 1 (ocean-bay interface closing MAR=1)

IBAY = 0 (shallow bay)

IHITE = 1 (regression surface not read in at time of high tide)

I1, I2, I3, I4, I5, I6 (refer to Fig. 3)

J1, J2, J3, J4, J5, J6 (refer to Fig. 3)

Vo, TPH, TPH1, A1, A2 = -90, 3138.7, -13420., 11.8872, 37.1856.

HIMIN, THT = 60.96, 13420.

TFLAT = 5640

CI, CC, CP, CH, CV = 1., 1., 1., 1., 1.

GR, FF, RR = 980., .00006, 1.

```
DX, DY, DZ = 160,000., 160,000., .25.
KH, KV = 10,000., 5.
BH, BV = 10,000, 5.
TI = 24.5
DTAUX(1) = -.37
DTAUY(1) = .15
DTA(1) = 31.7
DHS (1) = .00129
.
.
.
DT = 10
Next, IRUN = 1, LN = 300, LN1 = 1 (8:00am)
Next, IRUN = 1, LN = 300, LN1 = 1 (9:00am)
Next, IRUN = 1, LN = 420, LN1 = 1 (10:10am)
Next, IRUN = 1, LN = 1, LN1 =  1380 (2:00pm)
```

6.1.5 Program Execution Procedure (NASUM II)

This section describes the procedure by which the user executes the NASUM II program.

a) Input Parameters: The user must first follow the steps outlined in section 4.1, and become quite familiar with all the input parameters listed in section 5.1 and section 6.1.3.

b) First Run: In order to obtain surface heights and three-dimensional velocity and three-dimensional temperature, the main program FMAIN is executed. In FMAIN there are two tape units. One is a READ unit designated as Unit 7. The other is a STORE unit designated as unit 8. During the first run, ther is no need for unit 7, and unit 8 has to be provided to store results on a magnetic tape.

c) Run Continuation: For extending the results, the run has to be continued. The magnetic tape which was "unit 8" in the first run will now be read" Unit 7", for reading the previously stored results. Another magnetic tape is now to be provided as "unit 8" for storing the extended run results. The above procedure can be repeated until the results are obtained for the desired time. It is to be noted that for the first run IRUN=0, and for extended runs IRUN=1.

6.1.6 Sample Output

The output from the model sample run is listed as follows:

a) Parameters

b(Surface heights

c) Horizontal components of velocity

d) Vertical velocity component

e) Temperatures

```
HEILN                 $(1)  021715 023415     $(0)  140410 140614
                                              $(2)  BLANKSCOMMON
UVVEL                 $(1)  023416 030605     $(0)  140615 141215
                                              $(2)  BLANKSCOMMON
UVVELN                $(1)  030606 036142     $(0)  141216 141626
                                              $(2)  BLANKSCOMMON
WVEL                  $(1)  036143 037413     $(0)  141627 141741
                                              $(2)  BLANKSCOMMON
WVEL1                 $(1)  037414 040735     $(0)  141742 142013
                                              $(2)  BLANKSCOMMON
WVEL2                 $(1)  040736 043073     $(0)  142014 142105
                                              $(2)  BLANKSCOMMON
PRES                  $(1)  043074 043346     $(0)  142106 142160
                                              $(2)  BLANKSCOMMON
WW                    $(1)  043347 044353     $(0)  142161 142317
                                              $(2)  BLANKSCOMMON
TEMP                  $(1)  044354 046154     $(0)  142320 142523
                                              $(2)  BLANKSCOMMON
OLDUV                 $(1)  046155 046425     $(0)  142524 142570
                                              $(2)  BLANKSCOMMON
OLDHT                 $(1)  046426 046546     $(0)  142571 142621
                                              $(2)  BLANKSCOMMON
OLDT                  $(1)  046547 046701     $(0)  142622 142656
                                              $(2)  BLANKSCOMMON
ETT                   $(1)  046702 047024     $(0)  142657 142714
                                              $(2)  BLANKSCOMMON
PRPARA                $(1)  047025 047170     $(0)  142715 143052
                                              $(2)  BLANKSCOMMON
PRETA                 $(1)  047171 047266     $(0)  143053 143103
                                              $(2)  BLANKSCOMMON
PRUV                  $(1)  047267 047453     $(0)  143104 143154
                                              $(2)  BLANKSCOMMON
PRW                   $(1)  047454 047602     $(0)  143155 143221
                                              $(2)  BLANKSCOMMON
PRTEMP                $(1)  047603 047736     $(0)  143222 143263
                                              $(2)  BLANKSCOMMON
STORE                 $(1)  047737 050663     $(0)  143264 143356
                                              $(2)  BLANKSCOMMON

SYS$*RLIB$. LEVEL 7C-1
END OF COLLECTION - TIME 8.447 SECONDS

@XQT

 CI=     .1000000+01
 CM=     .1000000+01
 CV=     .1000000+01
 CP=     .1000000+01
 CC=     .1000000+01
 DX=     .1600000+06
 DY=     .1600000+06
 DZ=     .2500000+00
 DT=     .1000000+02
 TAUX=     .1130000+00
 TAUY=     .2500000+00
 TTOT=     .1342000+05
 GH=     .9630000+03
```

```
FF=       .6000000-04
RR=       .1129431+01
KH=       .1000000+05
KV=       .5000000+01

BH=       .1000000+05
BV=       .5000000+01
QRAD=       .0000000
TI=       .2450000+02
TTOT1=        .1000000+05
TA=       .3720000+02
HS=       .1360000-02


I =  1
WAVE-HEIGHT
            .0000000        .0000000        .0000000        .0000000        .0000000        .0000000        .0000000      -.1896096+02
          -.2526462+02    -.1764363+02    -.1869103+02

I =  2
WAVE-HEIGHT
            .0000000        .0000000        .0000000        .0000000        .0000000      -.8540977+01    -.1734145+02    -.9155718+01
          -.2233936+02    -.6626096+01    -.2122458+02

I =  3
WAVE-HEIGHT
            .0000000        .0000000        .0000000        .0000000        .0000000      -.1279902+02    -.2530216+02    -.1683669+02
          -.2526876+02    -.1709285+02    -.1469554+02

I =  4
WAVE-HEIGHT
            .0000000        .0000000        .0000000        .0000000        .0000000      -.1160079+02    -.2221517+02    -.1102045+02
          -.2181674+02    -.9829027+01    -.1948603+02

I =  5
WAVE-HEIGHT
            .0000000        .0000000        .0000000        .0000000      -.2687488+02    -.1667500+02    -.2575451+02    -.1587904+02
          -.2434985+02    -.1414262+02    -.2316542+02

I =  6
WAVE-HEIGHT
            .0000000        .0000000        .0000000        .0000000      -.2295670+02    -.1034982+02    -.2195090+02    -.1080090+02
          -.2139752+02    -.9770416+01    -.2260701+02

I =  7
WAVE-HEIGHT
            .0000000        .0000000        .0000000        .0000000      -.2475838+02    -.1551721+02    -.2417038+02    -.1518377+02
          -.2354714+02    -.1555955+02    -.2500125+02

I =  8
WAVE-HEIGHT
            .0000000        .0000000        .0000000      -.1027104+02    -.2178927+02    -.9791782+01    -.2163669+02    -.1022370+02
          -.2055770+02    -.1208865+02    -.2500125+02

I =  9
WAVE-HEIGHT
            .0000000        .0000000        .0000000      -.1540567+02    -.2478079+02    -.1499554+02    -.2361112+02    -.1408292+02
```

```
V-VELOCITY
        .0000000     .0000000     .0000000     .0000000     .0000000     .7366580+01  .1360999+02  .3740179+01
        .1626200+02  .1952644+01  .0000000

K=  1   I=  7
U-VELOCITY
        .0000000     .0000000     .0000000     .0000000    -.5661371+00  .8701912+01  .1453616+02  .2908626+02
        .4962534+02  .4995350+02  .0000000
V-VELOCITY
        .0000000     .0000000     .0000000     .0000000     .0000000     .7390385+01  .2093427+01  .3352377+02
        .7261625+01  .7815781+02  .7815781+02

K=  1   I=  8
U-VELOCITY
        .0000000     .0000000     .0000000     .0000000    -.9599721+01  .1284419+02 -.1784481+02  .2516354+02
        .2870513+02  .2025960+02  .0000000
V-VELOCITY
        .0000000     .0000000     .0000000     .0000000     .7899688+01  .1756063+02 -.4537699+01  .4386205+02
       -.2836532+00  .7815781+02  .7815781+02

K=  1   I=  9
U-VELOCITY
        .0000000     .0000000     .0000000     .4423000+01 -.8620792+01  .2529656+02 -.9834402+01  .2031744+02
        .4745767+02  .3166197+02  .0000000
V-VELOCITY
        .0000000     .0000000     .0000000     .0000000     .1203875+02  .2260893+02  .2170717+02  .3464679+02
        .4868098+02  .7815781+02  .7815781+02

K=  1   I= 10
U-VELOCITY
        .0000000     .0000000     .0000000    -.2600227+01 -.2287485+02  .9581270+01 -.2987218+02 -.3836036+01
        .2573546+02  .6537192+01  .0000000
V-VELOCITY
        .0000000     .0000000     .1014746+02  .8459224+01  .1741890+02  .2164779+02  .3003778+02  .2403834+02
        .5288198+02  .7815781+02  .7815781+02

K=  1   I= 11
U-VELOCITY
        .0000000     .2155363+01  .1244745+00  .3654298-01 -.1880044+02  .1487385+02 -.7240058+01 -.1901231+01
        .3071619+02  .7580710+01  .0000000
V-VELOCITY
        .0000000     .0000000     .1264938+02  .1769329+02  .1985914+02  .2770593+02  .2556618+02  .2884918+02
        .4268022+02  .7815781+02  .7815781+02

K=  1   I= 12
U-VELOCITY
        .0000000    -.4592051+01 -.1256923+02 -.5231626+01 -.4071285+02  .1008630+02 -.3948461+02 -.2137663+02
        .1263013+01 -.2306552+02  .0000000
V-VELOCITY
        .0000000     .0000000     .1513943+02  .1762959+02  .1631239+02  .1635323+02  .2736646+02  .2859461+02
        .3075564+02  .7815781+02  .7815781+02

K=  1   I= 13
U-VELOCITY
        .0000000     .0000000    -.6147593+01  .7632585+01 -.2071577+02  .1945293+02 -.2003917+02 -.1461359+02
        .1264269+01 -.1026899+02  .0000000
V-VELOCITY
```

```
K=  1     I =  9
 W -VELOCITY
          .0000000      .0000000      .0000000      .3492988-02   .3153560-02   .2776421-02   .2250031-02   .2101176-02
          .1974573-02   .2089752-02   .9782775-02
K=  1     I = 10
 W -VELOCITY
          .0000000      .3009129-02   .4560243-02   .4110550-02   .2907990-02   .3663823-02   .3610662-02   .2534645-02
          .3000594-02   .4043842-02   .1478153-01
K=  1     I = 11
 W -VELOCITY
          .0000000      .3277628-02   .3044439-02   .3072628-02   .2588939-02   .2342778-02   .2360796-02   .3168049-02
          .1624587-02   .3056898-02   .1164694-01
K=  1     I = 12
 W -VELOCITY
          .0000000      .1065260-01   .1607067-02   .3270246-02   .2509137-02   .7209129-03   .2564036-02   .2581194-02
          .2541044-02   .3046363-02   .1511435-01
K=  1     I = 13
 W -VELOCITY
          .0000000      .8151436-02   .2661427-02   .2880202-02   .3254401-02   .2629487-02   .3296794-02   .2601037-02
          .2725135-02   .2424482-02   .1100002-01
K=  1     I = 14
 W -VELOCITY
          .0000000      .0000000      .1811685-03   .4961285-02   .3032716-02   .2947769-02   .2601068-02   .2122584-02
          .3155266-02   .1775953-02   .1439950-01
K=  1     I = 15
 W -VELOCITY
          .0000000      .1105785-02   .4873958-02   .4775560-02   .3377179-02   .2663010-02   .3091018-02   .2127827-02
          .2966819-02   .3602575-02   .1369702-01
K=  1     I = 16
 W -VELOCITY
         -.4722691-02  -.1262246-02   .3156634-02   .2180497-02   .3309158-02   .2647301-02   .1655575-02   .1699278-02
          .1931192-02   .2488969-02   .1105699-01
K=  1     I = 17
 W -VELOCITY
          .1472839-01   .5333424-02   .2147616-02   .3040734-02   .3280763-02   .3921161-02   .2890130-02   .3089816-02
          .5953945-02   .3397438-02   .2416831-02
K=  1     I = 18
 W -VELOCITY
          .4737235-02   .2529672-03   .5262119-02   .1922037-02   .3780229-02   .2662869-02   .4022093-02   .3340506-02
          .2558575-02   .2329056-02   .6701294-02
K=  1     I = 19
 W -VELOCITY
          .4059941-02   .2792152-02   .2610294-02   .2024419-02   .2435638-02   .2208436-02   .2463631-02   .2279371-02
          .4446410-02   .1631194-02   .5946046-03
K=  1     I = 20
```

```
        .2520161+02    .2595073+02    .2601051+02

K=  1     I= 10
TEMPERATURE
        .0000000       .2734221+02    .2733197+02    .2666669+02    .2712409+02    .2572978+02    .2655099+02    .2668170+02
        .2617515+02    .2625217+02    .2632795+02

K=  1     I= 11
TEMPERATURE
        .0000000       .2733733+02    .2737013+02    .2693280+02    .2718791+02    .2653142+02    .2704911+02    .2707349+02
        .2613595+02    .2614964+02    .2616285+02

K=  1     I= 12
TEMPERATURE
        .0000000       .2841986+02    .2842696+02    .2823477+02    .2675550+02    .2818305+02    .2636028+02    .2722109+02
        .2568367+02    .2615415+02    .2613145+02

K=  1     I= 13
TEMPERATURE
        .0000000       .2661139+02    .2879942+02    .2851672+02    .2634614+02    .2795894+02    .2636244+02    .2650430+02
        .2570795+02    .2639211+02    .2632941+02

K=  1     I= 14
TEMPERATURE
        .0000000       .0000000       .2840187+02    .2841464+02    .2689815+02    .2647697+02    .2695362+02    .2668609+02
        .2568745+02    .2655999+02    .2653741+02

K=  1     I= 15
TEMPERATURE
        .0000000       .2885116+02    .2815197+02    .2768264+02    .2703379+02    .2581671+02    .2641289+02    .2700215+02
        .2733484+02    .2684851+02    .2693085+02

K=  1     I= 16
TEMPERATURE
        .2938051+02    .2954193+02    .2630793+02    .2760856+02    .2725777+02    .2532173+02    .2604870+02    .2803829+02
        .2611205+02    .2638505+02    .2631116+02

K=  1     I= 17
TEMPERATURE
        .2922190+02    .2922807+02    .2892668+02    .2754472+02    .2684028+02    .2506039+02    .2402145+02    .2919292+02
        .2570540+02    .2415605+02    .2411628+02

K=  1     I= 18
TEMPERATURE
        .2934978+02    .2935390+02    .2761890+02    .2707746+02    .2662792+02    .2732573+02    .2580546+02    .2271115+02
        .2544942+02    .2524984+02    .2517089+02

K=  1     I= 19
TEMPERATURE
        .2821745+02    .2822298+02    .2770765+02    .2730077+02    .2641872+02    .2726056+02    .2842338+02    .2602752+02
        .2568649+02    .3037807+02    .3035936+02

K=  1     I= 20
TEMPERATURE
        .2851732+02    .2652089+02    .2798062+02    .2689562+02    .2677064+02    .2635954+02    .2607883+02    .2690063+02
        .2832433+02    .2694248+02    .2362024+02
```

HORIZONTALLY STRETCHED MODEL

6.2. NASUM III (Horizontal Stretched Model):

6.2.1. Problem Statement:

Given the Hutchinson Island, St Lucie, Florida as an example application site, compute the three dimensional velocity and temperature distribution for the following discharge and mateorological conditions.

Discharge volume from the
Condensers of Power Plant } = 363,000 6 p.m.
Discharge Temperature = 35.0°C
Ambient Temperature = 25.0°C
Air Temperature = 30.0°C
Current = 2 cm/sec South

6.3. Calculation of Input Parameters:

In this section, the specification of grid system, reference quantities and calculation of discharge velocities chosen will be presented first followed by the actual calculation of input data as they appear in the main program.

6.3.1. Grid System

The remote sensing data and ground truth data was available for the Hutchinson Island site and it is used to determine the size of the domain. The domain selected was 2380 m x 2000 m. The domain has a variable depth and so a variable bottom topography is used. A horizontally stretched grid system as shown in Fig.( ) is used. This would give more resolution of the plume in the near field where the effects of the thermal discharge are predominant.

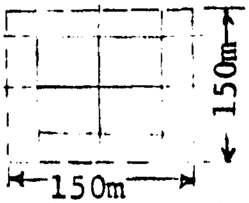
The grid system is also stretched in the vertical direction for ease of programming. In this way, same number of grid points can be used in shallow and deep regions of the basin.

6.3.2. Calculation of Discharge Velocity:

In the numerical model a 9 point discharge is chosen. The discharge velocity is calculated by balancing the mass as shown below.

For the numerical grid system, the mass into the domain = (Discharge Area x Velocity) = A x V

The grid size chosen at the discharge is the minimum grid and equal to 50 m. So the discharge area is equal to (150 x 150) m^2 as shown below



$$\therefore A = (150 \times 150) \times 10^4 \quad cm^2$$

$$V \times A = \text{Discharged volume (given)}$$

$$= 363{,}000 \text{ G.P.M}$$

$$V \times (150 \times 150 \times 10^4) = \frac{363{,}000 \times 0.0038 \times 10^6}{60} \quad \frac{cm^3}{sec}$$

$$V = 0.102 \text{ cm/sec}$$

$\therefore$Velocity at discharge or inlet velocity for the model is 0.102 cm/sec.

6.3.3. Reference Quantities:

The reference eddy viscosity and diffusivity are determined using the 4/3 power law as below

$$A_{ref} = 0.0025 \ (L)^{4/3}$$

Where A_{ref} is the reference eddy viscosity and L is the maximum length of the domain in centimeters.

$$A_{ref} = 0.0025\ (2380 \times 10^2)^{4/3}$$

$$\approx 40{,}000\ \mathrm{cm^2/sec}$$

For turbulent prandtl number of 1 ($Pr_t = \frac{A_{ref}}{B_{ref}}$)

B_{ref} (Eddy diffusivity) = 40,000 $\frac{cm^2}{sec}$

The vertical eddy viscosity and diffusivity chosen was 10 $\frac{cm^2}{sec}$

6.3.4. Calculation of Input Data as It Appears In the Main Program

Card No	Fortran Quantity
1	IRUN

IRUN = 0 for the first run and equal to 1 for later runs.

Card No	Fortran Quantity
2	LN

LN is the number of cycles required. It is always advised to run the program for 10 or 15 cycles and check how the model is running.

Card No	Fortran Quantity
3	KSTORE

If KSTORE is equal to zero the model sill store the results on the tape to be provided and if it is equal to 1 the model will not store results on the tape. For KSTORE equal to zero or 1 the model will print results at the end of the run,

Card No	Fortran Quantity
4	CI,CC,CH,CV,CP

The values of these quantities are equal to 1.0

Card No	Fortran Quantity
5	GR,FF,RR,HK

GR is gravity = 980.0 cm/sec

FF is coriolis term ≈ 0.0006

For small domain this terms is negligible and may be kept equal to 0.0 for all practical purposes

RR is density = 1.0

$$HK = \frac{h}{PC_pKV}$$

h = 1200 BTU/day °F - ft = 0.00678 Ca/sec,°C cm

$$HK = \frac{0.00678}{1.0x1.0x1.0} = 0.000678$$

Card No	Fortran Quantity
6	DX,DY,DZ

DX,DY,DZ are the minimum grid sizes chosen and they are

DX = 5000.0

DY = 5000.0

$$DZ = \frac{1}{KN-1} = \frac{1}{5-1} = \frac{1}{4} = 0.25$$

When KN is the number of grids in the vertical direction

Card No	Fortran Quantity
7	KH,KV,BH,BV

These are reference horizontal and vertical eddy viscosities and diffusivities. These are calculated in section 6.3.3.

KH = 40000.0

KV = 10.0

BH = 40000.0

BV = 10.0

Card No	Fortran Quantity
8	DELX,DELY,DEEX,DEEY,EEEX,EEEY

DELX and DELY are minimum grid sizes chosen = 5000 cm

In order to determine DEEX,DEEY,EEEX,EEEY, another main program "CONST" has to be run twice. The input cards needed for the program "CONST" are

First time XB,A,DX,AN

Second time YB,B,DY,BN

Where XB = X boundary = 238000.0 cm

A = a constant = 38,000.0

DX = DELTA X = 5000.0 cm

AN = Number of grids in x direction = 20

Similarly YB = 200000.0 cm

B = 100000.0

DY = 5000.0 cm

BN = 20

The output of this main program will give the constants C1,D and C1,D

First C1,D are equal to EEEX and DEEX

Second C1,D are equal to EEEY and DEEEY.

For the sample problem they are equal to

DEEX = 29333.03

DEEY = 47503.21

EEEX = 22947.29

EEEY = 20957.83

Card No	Fortran Quantity
9	TAUX,TAUY

For the sample problem the effect of wind is neglected and they are equal to 0.0,0.0. If the effects wind are to be considered, they can be easily calculated as explained in Appendix A.

Card No	Fortran Quantity
10	TAIR

TAIR, the air temp = 30.0°C

Card No	Fortran Quantity
11	DT

The value of DT used in the sample problem is 5 sec. In general it is always advised to start with a small value and increase it to the point when the model would not go unstable.

Note: The sample problem presented is a simplified version of the Hutchinson Island discharge problem.

6,3.5, Program Execution Procedure (NASUM III)

This section describes the procedure by which the User executes the NASUM III program,

a) Input Parameter:

The User must first follow the steps outlined in the sample problem and become quite familiar with all the input parameters,

b) First Run:

In order to obtain three-dimensional velocity and three-dimensional temperature, the main program TMAIN3 is executed. In TMAIN3 there are two tape units. One is a READ unit designated as Unit 7. During the first run, there is no need for unit 7, and unit 8 has to be provided to store results on a magnetic tape.

c) Run Continuation:

For extending the results, the run has to be continued. The magnetic tape which was "unit 8" in the first run will now be read "unit 7", for reading the previously stored results. Another magnetic tape is now to be provided as "unit 8" for storing the extended run results. The above procedure can be repeated until the results are obtained for the desired time. It is to be noted that for the first run IRUN = 0, and for extended runs IRUN = 1.

6.3.6. Sample Input

Card No	Symbol	Value & Format
1	IRUN	b̸ b̸ b̸ b̸ 0
2	LN	b̸ b̸ 700
3	KSTORE	b̸ b̸ b̸ b̸ 0
4	CI,CC,CP,CH,CV	1.0,1.0,1.0,1.0,1.0
5	GR,FF,RR,HK	980.0,0.00006,1.0,0.000678
6	DX,DY,DZ	5000.0,5000.0,0.25
7	KH,KV,BH,BV	40000.0,10.0,40000.0,10.0
8	DELX,DELY,DEEX,	5000.0,5000.0,29333.03,
	DEEY,EEEX,EEEY	47503.21,22947.29,20957.83
9	TAUX,TAUY	0.0,0.0
10	TAIR	30.0
11	DT	5.0

6.3.7. Sample Output

Some of the output from the model sample run are listed as follows:

a) Parameters

b) Surface (K=1) horizontal velocities, u and v.

c) Vertical velocity, Ω, at K=2

d) Surface temperatures (J=1 at left-hand side)
(J=JN at right-hand side)
I=1 at top
I=IN at bottom

The surface isotherms obtained after 1 hour of simulation are shown in Fig.(11).

```
          .2000000+01     .1545517+01     .1759030+01     .1774168+01     .1728563+01     .1731295+01     .1763643+01     .1773118+01
          .1811704+01     .1656779+01     .1896564+01     .1910914+01     .1889646+01     .1843848+01     .1791741+01     .1752052+01
          .1730221+01     .1726093+01     .1724963+01     .1724963+01

K=  1   I=   6
U-VELOCITY
          .3859683+00     .3859683+00     .3777352+00     .3737937+00     .3730599+00     .3743002+00     .3789361+00     .3892725+00
          .4089655+00     .4303837+00     .4486609+00     .4528125+00     .4388481+00     .4167932+00     .3957166+00     .3613283+00
          .3746219+00     .3726367+00     .3722797+00     .3722797+00
V-VELOCITY
          .2000000+01     .1697179+01     .1825772+01     .1804027+01     .1799613+01     .1805433+01     .1827792+01     .1877697+01
          .1963172+01     .2076142+01     .2161481+01     .2184455+01     .2117109+01     .2010683+01     .1968917+01     .1839366+01
          .1506944+01     .1797329+01     .1795598+01     .1795598+01

K=  1   I=   7
U-VELOCITY
          .3962599+00     .3962599+00     .3875602+00     .3643059+00     .3837645+00     .3857023+00     .3928139+00     .4100549+00
          .4446257+00     .4492876+00     .5366941+00     .5431659+00     .5005581+00     .4541865+00     .4178398+00     .3957368+00
          .3861061+00     .3833617+00     .3828788+00     .3828788+00
V-VELOCITY
          .2000000+01     .1934309+01     .1877936+01     .1654613+01     .1851303+01     .1860526+01     .1898846+01     .1976075+01
          .2142628+01     .2408703+01     .2599210+01     .2620639+01     .2415021+01     .2191283+01     .2015611+01     .1939035+01
          .1862449+01     .1849161+01     .1846821+01     .1846821+01

K=  1   I=   8
U-VELOCITY
          .4021191+00     .4021191+00     .3941110+00     .3910665+00     .3907086+00     .3932578+00     .4027516+00     .4278091+00
          .4856136+00     .6316023+00     .7091531+00     .7152359+00     .5782492+00     .4908179+00     .4359409+00     .4053369+00
          .3937297+00     .3903276+00     .3897365+00     .3897365+00
V-VELOCITY
          .2000000+01     .1957667+01     .1934778+01     .1887256+01     .1884846+01     .1897045+01     .1942869+01     .2063817+01
          .2347581+01     .3047144+01     .3421393+01     .3450842+01     .2790002+01     .2368155+01     .2133270+01     .1959829+01
          .1899339+01     .1862840+01     .1879986+01     .1879986+01

K=  1   I=   9
U-VELOCITY
          .4057495+00     .4057495+00     .3982729+00     .3954417+00     .3951667+00     .3980469+00     .4068385+00     .4350759+00
          .5066658+00     .6849653+00     .7823523+00     .7911978+00     .6222743+00     .5130446+00     .4470165+00     .4124351+00
          .3985821+00     .3946117+00     .3941632+00     .3941632+00
V-VELOCITY
          .2000000+01     .1971946+01     .1924340+01     .1908209+01     .1906390+01     .1920205+01     .1972297+01     .2113417+01
          .2464349+01     .3304657+01     .3774657+01     .3617395+01     .3002522+01     .2475497+01     .2156797+01     .1989776+01
          .1922792+01     .1904529+01     .1901390+01     .1901390+01

K=  1   I=  10
U-VELOCITY
          .4085524+00     .4085524+00     .4010060+00     .3983351+00     .3980718+00     .4007681+00     .4107968+00     .4376726+00
          .5019294+00     .6624100+00     .7506443+00     .7588659+00     .6058703+00     .5073906+00     .4467816+00     .4148562+00
          .4013417+00     .3977452+00     .3971243+00     .3971243+00
V-VELOCITY
          .2000000+01     .1960919+01     .1937163+01     .1922136+01     .1920436+01     .1933377+01     .1981790+01     .2111518+01
          .2421603+01     .3155934+01     .3621737+01     .3661547+01     .2923429+01     .2446259+01     .2155701+01     .1999555+01
          .1936143+01     .1916733+01     .1915725+01     .1915725+01

K=  1   I=  11
U-VELOCITY
          .4098336+00     .4098336+00     .4032376+00     .4007183+00     .4003613+00     .4022570+00     .4090276+00     .4251279+00
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        -.5986951-06  -.6065294-06  -.5999640-06  -.5795549-06  -.5514443-06  -.5271942-06  -.5172344-06  -.5216362-06
        -.5303639-06  -.5355279-06  -.5370959-06  -.5370959-06

K=  2     I =  3
 W -VELOCITY
        -.2339585-05  -.2339585-05  -.5822766-06  -.5986260-06  -.6637482-06  -.7096751-06  -.7672427-06  -.8407868-06
        -.9093709-06  -.9423040-06  -.9190950-06  -.8429713-06  -.7421081-06  -.6624772-06  -.6334571-06  -.6469847-06
        -.6708138-06  -.6841850-06  -.6881060-06  -.6881060-06

K=  2     I =  4
 W -VELOCITY
        -.2521598-05  -.2521598-05  -.7143459-06  -.6528320-06  -.7089765-06  -.7891596-06  -.9341541-06  -.1146161-05
        -.1359309-05  -.1457493-05  -.1358190-05  -.1087410-05  -.7684996-06  -.5669878-06  -.5315889-06  -.5977216-06
        -.6693902-06  -.7042214-06  -.7136579-06  -.7136579-06

K=  2     I =  5
 W -VELOCITY
        -.2603294-05  -.2603294-05  -.7760259-06  -.6495495-06  -.7052560-06  -.8500963-06  -.1169251-05  -.1698952-05
        -.2269829-05  -.2545871-05  -.2208031-05  -.1391721-05  -.5407673-06  -.1603215-06  -.2040348-06  -.4176738-06
        -.5846590-06  -.6622075-06  -.6806713-06  -.6806713-06

K=  2     I =  6
 W -VELOCITY
        -.2620122-05  -.2620122-05  -.7929078-06  -.6174251-06  -.6820693-06  -.9239129-06  -.1528038-05  -.2650884-05
        -.4122078-05  -.4902348-05  -.3889488-05  -.1675533-05   .3564357-06   .7958045-06   .4072236-06  -.1231863-06
        -.4610948-06  -.5905427-06  -.6210384-06  -.6210384-06

K=  2     I =  7
 W -VELOCITY
        -.2350354-05  -.2350354-05  -.5274738-06  -.3193416-06  -.3989619-06  -.7550837-06  -.1744114-05  -.3884569-05
        -.7459896-05  -.9650002-05  -.6618491-05  -.9060569-06   .3579620-05   .3047501-05   .1640463-05   .5252009-06
        -.5568851-07  -.2564375-06  -.3006552-06  -.3008552-06

K=  2     I =  8
 W -VELOCITY
        -.2106413-05  -.2106413-05  -.2813387-06  -.5652473-07  -.1536991-06  -.6161365-06  -.2026971-05  -.5695058-05
        -.1534233-04  -.1641256-03  -.1541334-03  -.1392077-03   .1252752-04   .6503024-05   .3044076-05   .1146569-05
         .3010527-06   .3159564-07  -.2542270-07  -.2542270-07

K=  2     I =  9
 W -VELOCITY
        -.2096593-05  -.2096593-05  -.2743979-06  -.4365710-07  -.1448524-06  -.6561055-06  -.2253665-05  -.6493865-05
        -.1723106-04  -.1652631-03  -.1524090-03  -.1341178-03   .1699834-04   .8630913-05   .3854697-05   .1434547-05
         .3774375-06   .6328812-07  -.1476744-08  -.1476744-06

K=  2     I = 10
 W -VELOCITY
        -.2076426-05  -.2076426-05  -.2633086-06  -.2693833-07  -.1144627-06  -.5703054-06  -.1963840-05  -.5573972-05
        -.1474647-04  -.1608121-03  -.1483602-03  -.1318735-03   .1688310-04   .8440196-05   .3778330-05   .1391255-05
         .3836446-06   .8116115-07   .1797993-07   .1797993-07

K=  2     I = 11
 W -VELOCITY
        -.2205004-05  -.2205004-05  -.4193062-06  -.1757205-06  -.2264984-06  -.5150626-06  -.1270631-05  -.2717786-05
        -.4538576-05  -.3311065-05   .5895869-06   .5361433-05   .6255967-05   .4368697-05   .2282364-05   .6438079-06
         .1455406-06  -.6389181-07  -.1329028-06  -.1329028-06
```

```
TEMPERATURE AT K =    1

26.49 26.49 25.78 25.51 25.43 25.43 25.46 25.51 25.57 25.62 25.66 25.68 25.66 25.61 25.54 25.46 25.41 25.40 25.39 25.39
25.49 26.49 25.78 25.51 25.43 25.43 25.46 25.51 25.57 25.62 25.66 25.68 25.66 25.61 25.54 25.46 25.41 25.40 25.39 25.39
26.35 26.35 25.73 25.43 25.42 25.44 25.51 25.62 25.75 25.87 25.96 25.99 25.96 25.85 25.68 25.52 25.42 25.38 25.37 25.37
26.21 26.21 25.69 25.47 25.42 25.48 25.62 25.86 26.12 26.38 26.58 26.65 26.57 26.33 25.98 25.65 25.45 25.38 25.37 25.37
26.10 26.10 25.67 25.48 25.45 25.55 25.77 26.20 26.65 27.18 27.59 27.71 27.52 27.05 26.42 25.85 25.52 25.41 25.38 25.38
26.04 26.04 25.67 25.51 25.50 25.63 25.66 26.64 27.65 28.64 29.17 29.27 28.61 27.96 26.96 26.11 25.62 25.46 25.42 25.42
26.04 26.04 25.68 25.56 25.57 25.76 26.17 27.31 28.96 30.90 31.52 31.51 30.42 29.00 27.54 26.38 25.74 25.53 25.48 25.48
26.05 26.05 25.67 25.60 25.63 25.87 26.44 27.94 30.47 35.00 35.00 35.00 32.19 29.57 28.04 26.61 25.84 25.59 25.54 25.54
26.04 26.04 25.52 25.61 25.66 25.92 26.56 28.20 30.88 35.00 35.00 35.00 32.77 30.43 28.30 26.72 25.89 25.62 25.57 25.57
26.01 26.01 25.52 25.60 25.66 25.89 26.47 27.99 30.55 35.00 35.00 35.00 32.42 30.16 28.16 26.67 25.88 25.62 25.57 25.57
25.93 25.93 25.63 25.58 25.60 25.77 26.10 27.09 26.47 30.04 30.72 30.85 30.07 28.84 27.49 26.39 25.78 25.58 25.54 25.54
25.84 25.84 25.59 25.54 25.55 25.64 25.84 26.13 26.75 27.39 27.81 27.96 27.76 27.27 26.61 26.00 25.65 25.53 25.50 25.50
25.78 25.78 25.38 25.50 25.52 25.56 25.69 25.77 25.96 26.15 26.29 26.36 26.32 26.17 25.94 25.71 25.56 25.51 25.50 25.50
25.72 25.72 25.37 25.50 25.52 25.53 25.55 25.58 25.63 25.67 25.70 25.72 25.71 25.68 25.63 25.57 25.53 25.52 25.52 25.52
25.67 25.67 25.36 25.50 25.53 25.53 25.54 25.54 25.55 25.55 25.56 25.56 25.56 25.55 25.55 25.54 25.53 25.53 25.53 25.53
25.63 25.63 25.35 25.50 25.53 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54
25.59 25.59 25.34 25.50 25.53 25.54 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55
25.55 25.55 25.33 25.49 25.53 25.54 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55
25.53 25.53 25.33 25.50 25.54 25.55 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56
25.53 25.53 25.33 25.50 25.54 25.55 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56
```

7.1 MAIN PROGRAM FOR NASUM II

```
C      FMAIN
C***********************************************************************
C      THIS PROGRAM CALCULATES THE SURFACE ELEVATIONS, CIRCULATION AND
C      TEMPERATURE DISTRIBUTION FOR THE THREE-DIMENSIONAL FREE SURFACE
C      FAR-FIELD MODEL
C***********************************************************************
       PARAMETER IN=34,JN=11,KN=5
       REAL KH,KV
       DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),
      CHT(IN,JN),HI(IN,JN),ETA(IN,JN),HTE(IN,JN),
      CH(IN,JN,KN),G(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
      CHX(IN,JN),HY(IN,JN),MAR(IN,JN),HTD(IN,JN),
      CRO(IN,JN,KN),P(IN,JN,KN),HDUM(IN,JN)
      C,T(IN,JN,KN),TN1(IN,JN,KN),WZ(IN,JN,KN)
      C,II(IN),DTA(13),DHS(13),DTAUX(13),DTAUY(13)
       READ 1,IRUN
       READ 1,LN
       READ 1,LN1
       READ 1,M
       READ 1,INLET
       READ 1,IBAY
       READ 1,IH1TE
1      FORMAT (I5)
       READ 11,I1,I2,I3,I4,I5,I6
       READ 11,J1,J2,J3,J4,J5,J6
11     FORMAT(6I5)
       READ 2,V0,TPH,TPH1,A1,A2
       READ 2,HTMIN,THT
       READ 2,TFLAT
       READ 2, CI,CC,CF,CH,CV
       READ 2, GR,FF,RR
       READ 2, DX,DY,DZ
       READ 2, KH,KV
       READ 2,BH,BV
       READ 2,TI
       READ 2,(DTAUX(I),I=1,13)
       READ 2,(DTAUY(I),I=1,13)
       READ 2,(DTA(I),I=1,13)
       READ 2,(DHS(I),I=1,13)
       DO 100 I=1,13
       DHS(I)=DHS(I)/100.
100    CONTINUE
       READ 2,DT
2      FORMAT(   )
       IF(IRUN.GT.0) GO TO 4
       CALL READ2(IN,JN,MAR,HI,HX,HY,DX,DY)
       CALL INITIA(IN,JN,KN,ETA,HT,HI,HTD,HTE,II,U,V,W,RO,P,GR,RR,DZ,D,
      CE,H,G,TI,T,TN1)
       TTOT=0.0
       TTOT1=0.0
       TAUX=0.0
       TAUY=0.0
       GO TO 6
4      CONTINUE
       CALL READ1(IN,JN,KN,U,V,W,HI,HT,HTD,HX,HY,MAR,ETA,P,RO,CI,
      CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TTOT1,WZ)
```

```
6     CONTINUE
      DO 5 L=1,LN
      IF(TTOT.GE.THT) GO TO 1000
      TTOT=TTOT+DT
      TT=TTOT
      IF(M.EQ.1) GO TO 33
      CALL TIDE(IN,JN,TT,L,DT,HTD,HI,HTE,INLET,I1,I2,J1,J3,J4,I5,TPH1
     C,A1,A2)
      GO TO 34
33    CALL VEL(IN,JN,KN,U,V,H,G,D,E,TT,VO,TPH,I1,I2,I5,J1,J3,J4,INLET)
34    CONTINUE
      CALL DATA(TAUX,TAUY,TT,TA,HS,DTAUX,DTAUY,DTA,DHS,TFLAT)
      ITTOT=TTOT
      IF(L.GT.1.OR.ITTOT.GT.0) GO TO 50
      CALL HEIGHT(I,J,K,IN,JN,KN,MAR,U,V,HT,HTD,DZ,DT,DX,DY,HDUM,M,I1,
     CI2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
      CALL UVVEL(IN,JN,KN,U,V,H,G,DX,DY,DZ,W,TAUX,TAUY,DT,HT,
     CHX,HY,ETA,P,MAR,CI,CC,CP,CH,CV,KH,KV,GR,RR,FF,HTD,RO,T,I1,I2,I3,
     CI4,I5,I6,J1,J2,J3,J4,J5,J6,M)
      CALL WVEL(IN,JN,KN,H,G,W,HTD,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
      CALL PRES(IN,JN,KN,HTD,RO,GR,P,DZ)
      CALL ETT(IN,JN,HTD,HI,MAR,ETA)
      GO TO 80
50    CONTINUE
      CALL HEILN(IN,JN,KN,MAR,H,G,HTD,HT,HTE,DZ,DT,DX,DY,HDUM,M,I1,I2,
     CI3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
      CALL UVVELN(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
     CHT,HTD,HTE,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,CP,CC,CI,CH,CV,RO,T,I1
     C,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6,M)
      CALL WVEL(IN,JN,KN,D,E,W,HTE,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
      CALL PRES(IN,JN,KN,HTE,RO,GR,P,DZ)
      CALL ETT(IN,JN,HTE,HI,MAR,ETA)
      CALL OLDUV(IN,JN,KN,U,V,H,G,D,E)
      CALL OLDHT(IN,JN,HTE,HTD,HT)
80    CONTINUE
5     CONTINUE
1000  CONTINUE
      DO 15 LL=1,LN1
      TT=TTOT1+TTOT
      IF(TT.LT.THT) GO TO 3000
      IF(IBAY.EQ.1) GO TO 5002
      DO 211 I=1,IN
      DO 211 J=1,JN
      IF(ETA(I,J).LT.0.0) GO TO 65
      GO TO 211
65    CONTINUE
      IHI=HI(I,J)
      IHTMIN=HTMIN
      IF(IHI.EQ.IHTMIN) GO TO 4000
      GO TO 4001
4000  CONTINUE
      HTM=HTMIN+30.48
      HI(I,J)=HTM
      HT(I,J)=HI(I,J)+ETA(I,J)
      HTD(I,J)=HT(I,J)
      HTE(I,J)=HT(I,J)
4001  CONTINUE
```

```
      A=-30.48
      HTM=HTMIN+30.48
      IF(ETA(I,J).LE.A.AND.HI(I,J).LE.HTM) GO TO 311
      GO TO 211
 311  CONTINUE
      HI(I,J)=HTM+30.48
      HT(I,J)=ETA(I,J)+HI(I,J)
      HTD(I,J)=HT(I,J)
      HTE(I,J)=HT(I,J)
 211  CONTINUE
 5002 CONTINUE
      IF(M.EQ.1) GO TO 330
      CALL TIDE(IN,JN,TT,LL,DT,HTD,HI,HTE,INLET,I1,I2,J1,J3,J4,I5,TPH1
     C,A1,A2)
      GO TO 340
 330  CALL VEL(IN,JN,KN,U,V,H,G,D,E,TT,VO,TPH,I1,I2,I5,J1,J3,J4,INLET)
 340  CONTINUE
      CALL DATA(TAUX,TAUY,TT,TA,HS,DTAUX,DTAUY,DTA,DHS,TFLAT)
      ITT=TT
      ITHT=THT
      IF(ITT.EQ.ITHT) GO TO 2000
      TTOT1=TTOT1+DT
 500  CONTINUE
      CALL HEILN(IN,JN,KN,MAR,H,G,HTD,HT,HTE,DZ,DT,DX,DY,HDUM,M,I1,
     CI2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
      GO TO 2001
 2000 CONTINUE
      CALL TIDAL(IN,JN,KN,ETA,HI,HT,HTD,HTE,II,T,IHITE)
      TTOT1=TTOT1+DT
      IF(IHITE.EQ.1) GO TO 500
 2001 CONTINUE
      CALL UVVELN(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
     CHT,HTD,HTE,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,CP,CC,CI,CH,CV,RO,T,I1,
     CI2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6,M)
      CALL WVEL(IN,JN,KN,D,E,W,HTE,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
      CALL PRES(IN,JN,KN,HTE,RO,GR,P,DZ)
      CALL ETT(IN,JN,HTE,HI,MAR,ETA)
      CALL OLDUV(IN,JN,KN,U,V,H,G,D,E)
      CALL OLDHT(IN,JN,HTE,HTD,HT)
      CALL TEMP(IN,JN,KN,T,H,G,W,DX,DY,DZ,DT,HT,BH,BV,MAR,TN1,HTD,QRAD,
     CRR,HS,TA,RO)
      CALL OLDT(IN,JN,KN,T,TN1)
 15   CONTINUE
 3000 CONTINUE
      CALL PRPARA(CI,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,
     CKH,KV,BH,BV,QRAD,TI,TTOT1,TA,HS)
      CALL PRETA(I,J,IN,JN,ETA)
      CALL PRUV(I,J,K,IN,JN,KN,H,G)
      CALL WW(IN,JN,KN,HT,HTD,ETA,H,G,W,WZ,MAR,DX,DY,DZ,DT,HX,HY)
      CALL PRW(IN,JN,KN,WZ)
      CALL PRTEMP(I,J,K,IN,JN,KN,T)
      CALL STORE(IN,JN,KN,U,V,W,HI,HT,HTD,HX,HY,MAR,ETA,P,RO,CI,
     CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TTOT1,WZ)
      STOP
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE PROGRAMS FOR NASUM II

7.1.1 READ 2

This subroutine is used by specifying IRUN=0. First, the two-dimensional MAR matrix, MAR (I,J) is read in from data cards (a sample problem will illustrate this in Section 6.1.

The MAR numbering system is used for distinguishing between spatial differencing of the terms of the system of governing equations in the interior of the domain of solution, on the boundaries, and outside the domain. Points outside the domain are assigned a value MAR=0, and calculations are not performed. The MAR matrix, as will be shown in a sample problem, is constructed by the user by first establishing a grid system which closely follows the geometry of the application site. Then the MAR numbering system is specified as follows:

MAR (I,J) =0 points outside domain

MAR (I,J) =1 upper horizontal boundary

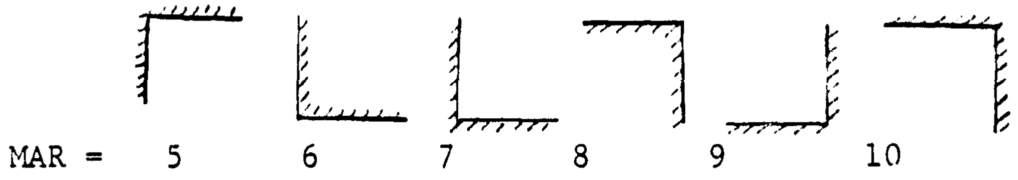
MAR (I,J) =2 lower horizontal boundary

MAR (I,J) =3 left vertical boundary

MAR (I,J) =4 right vertical boundary

MAR (I,J) =5 through MAR (I,J) = 10 are boundary corners and are specified below:

MAR (I,J) =11interior of domain.



Next, this subroutine reads in from data cards the two-dimensional matrix, HI(I,J), which specifies the depth (in feet) below the mean water level at each grid point. Then the depths are converted into centimeters for calculation of the bottom gradients, HX(I,J) and HY(I,J) in the x and y directions, respectively. Note, that calculation of the bottom gradients is performed by using central differencing in the interior and the point single-sided differencing on the boundary. Again, for MAR=6 and MAR=8 central differencing is used, since these particular corners may be treated as being interior points.

```
C*******************************************************************************
C      THIS SUBROUTINE READS IN DATA FOR THE APPLICATION SITE DOMAIN
C      AND ACTUAL BOTTOM TOPOGRAPHY
C*******************************************************************************
       SUBROUTINE READ2(IN,JN,MAR,HI,HX,HY,DX,DY)
       DIMENSION MAR(IN,JN),HI(IN,JN),HX(IN,JN),HY(IN,JN)
       DO 10 I=1,IN
10     READ 1,(MAR(I,J),J=1,JN)
       DO 99 I=1,IN
99     READ 1,(HI(I,J),J=1,JN)
1      FORMAT(   )
       DO 15 I=1,IN
       DO 15 J=1,JN
       HI(I,J)=30.48*HI(I,J)
15     CONTINUE
       DO 50 I=1,IN
       DO 50 J=1,JN
       IF(MAR(I,J).EQ.0) GO TO 50
       IF(MAR(I,J).EQ.1) GO TO 31
       IF(MAR(I,J).EQ.2) GO TO 32
       IF(MAR(I,J).EQ.3) GO TO 33
       IF(MAR(I,J).EQ.4) GO TO 34
       IF(MAR(I,J).EQ.5) GO TO 35
       IF(MAR(I,J).EQ.6) GO TO 36
       IF(MAR(I,J).EQ.7) GO TO 37
       IF(MAR(I,J).EQ.8) GO TO 38
       IF(MAR(I,J).EQ.9) GO TO 39
       IF(MAR(I,J).EQ.10) GO TO 40
       HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
       HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
       GO TO 50
31     CONTINUE
       HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
       HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
       GO TO 50
32     CONTINUE
       HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
       HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
       GO TO 50
33     CONTINUE
       HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
       HY(I,J)=(HT(I,J+1)-HI(I,J-1))/(2.*DY)
       GO TO 50
34     CONTINUE
       HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
       HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
       GO TO 50
35     CONTINUE
       HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
       HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(1,J-1))/(2*DY)
       GO TO 50
36     CONTINUE
       HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
       HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
       GO TO 50
37     CONTINUE
```

[redacted]______ INTENTIONALLY BLANK

```
             HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
             HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
             GO TO 50
      38     CONTINUE
             HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
             HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
             GO TO 50
      39     CONTINUE
             HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
             HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
             GO TO 50
      40     CONTINUE
             HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
             HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
      50     CONTINUE
             RETURN
             END
```

7.1.2 INITIA

This subroutine is used by specifying IRUN =0. The initial values of η, H, u, v, Ω, T, P and ρ are specified as was outlined in section 4.1.

$\eta = 0$

$u = 0$

$v = 0$

$\Omega = 0$

*$T = T_{unif} =$ const.

$P = P_o + \rho g H \sigma$

$\rho = P_{unif}$

In terms of the symbols used in the program, we have:

```
ETA(I,J) = 0
HT(I,J)  = HI(I,J)
HTD(I,J) = HI(I,J)
HTE(I,J) = HI(I,J)
U(I,J,K) = 0
H(I,J,K) = 0
D(I,J,K) = 0
V(I,J,K) = 0
G(I,J,K) = 0
E(I,J,K) = 0
W(I,J,K) = 0
T(I,J,K) =TI
TN1(I,J,K)=TI
RO(I,J,K) =RR
P(I,J,K)  =RR*GR*HT(I,J)*(K-1)*DZ
```

*NOTE: The energy equation is only coupled to the system of governing equations after the IR data base is inputted at 10:10pm for 4/15/75 into the program, and therefore an initial value of T is not actually required. However, if no IR data is available an isothermal bay may be assumed initially and the coupling to the energy equation initially.

```
C     THIS PROGRAM INITIALIZES VARIABLES FOR CONSTANT DENSITY MODEL
C
C***************************************************************************
C     THIS SUBROUTINE INITIALIZES THE VARIABLES FOR THE VARIABLE DENSITY M
C***************************************************************************
      SUBROUTINE INITIA(IN,JN,KN,ETA,HT,HI,HTD,HTE,II,U,V,W,RO,P,GR,RR,
     CDZ,D,E,H,G,TI,T,TN1)
      DIMENSION ETA(IN,JN),HT(IN,JN),HI(IN,JN),HTD(IN,JN),HTE(IN,JN),
     CII(IN),U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),W(IN,JN,KN)
     C,RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),P(IN,JN,KN),T(IN,JN,KN),
     CTN1(IN,JN,KN)
      DO 20 I=1,IN
      DO 20 J=1,JN
      ETA(I,J)=0.0
      HT(I,J)=HI(I,J)+ETA(I,J)
      HTD(I,J)=HT(I,J)
      HTE(I,J)=HT(I,J)
20    CONTINUE
      DO 8 I=1,IN
      DO 8 J=1,JN
      DO 8 K=1,KN
      U(I,J,K)=0.
      H(I,J,K)=0.
      D(I,J,K)=0.
      V(I,J,K)=0.
      G(I,J,K)=0.
      E(I,J,K)=0.
      W(I,J,K)=0.
      T(I,J,K)=TI
      TN1(I,J,K)=TI
      RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
8     CONTINUE
      DO 25 I=1,IN
      DO 25 J=1,JN
      DO 21 K=1,KN
      RR=RO(I,J,K)
      P(I,J,K)=GR*HT(I,J)*RR*(K-1)*DZ
21    CONTINUE
25    CONTINUE
      RETURN
      END
```

7.1.3 READ 1

This subroutine is used by specifying IRUN=1. The results of a previous run are read in from a magnetic tape for the purpose of running the program over a long period of time in segments. The system variables η, H, u, v, Ω, T. P and ρ are read in as well as physical and numerical parameters, as follows:

HI(I,J) = depths at each grid points (in cm)

HX(I,J) = bottom gradient in x-direction

HY(I,J) = bottom gradient in y-direction

MAR(I,J) = domain numbering system

CI, CC, CH, CV, CP = constants always = 1

DX = horizontal grid size in z-direction (in cm)

DY = horizontal grid size in y-direction (in cm)

DZ = vertical grid size = ΔZ/HI(I,J)

*DT = time step (in seconds)

TAUX = wind stress component in x-direction (dynes/cm^2)

TAUY = wind stress component in y-direction (dynes/cm^2)

TTOT = total run time without energy eouation (in seconds)

TTOT1 = total run time with energy equation (in seconds)

* NOTE: The determination of Δt is presented in the sample problem in section 6.1.

```
C
C****************************************************************************
C     THIS SUBROUTINE READS IN(FROM MAGNETIC TAPE) VALUES FOR THE VARIABLE.
C     AND PHYSICAL AND NUMERICAL PARAMETERS FROM A PREVIOUS COMPUTER RUN
C     FOR THE VARIABLE DENSITY MODEL
C****************************************************************************
      SUBROUTINE READ1(IN,JN,KN,U,V,W,HI,HT,HTD,HX,HY,MAR,ETA,P,RO,CI,
     CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TTOT1,WZ)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),P(IN,JN,KN),
     CHI(IN,JN),HT(IN,JN),HTD(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),
     CETA(IN,JN),RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN)
     C,T(IN,JN,KN),WZ(IN,JN,KN)
      READ (7)  (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((H(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((G(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((P(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C((HTD(I,J),J=1,JN),I=1,IN),
     C((HTE(I,J),J=1,JN),I=1,IN),
     C((HI(I,J),J=1,JN),I=1,IN),
     C((HX(I,J),J=1,JN),I=1,IN),
     C((HY(I,J),J=1,JN),I=1,IN),
     C((MAR(I,J),J=1,JN),I=1,IN),
     C((HT(I,J),J=1,JN),I=1,IN),
     C((ETA(I,J),J=1,JN),I=1,IN),
     C(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((WZ(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     CCI,CC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,TTOT1
      RETURN
      END
```

7.1.4 TIDE (M=2)

This subroutine specifies the tide height at the ocean-bay opening as a function of time (i.e. TTOT or TTOT + TTOT1) in the form:

$\eta_o (t) = A_1 + A_2 \cos \omega(t+t\phi)$

or in terms of the program symbols:

ETA = A1 + A2 *cos ((TT + TPH1)*(6.23/12.15 *3600))

where A1 and A2 are read in from data cards, TT = TTOT or =TTOT + TTOT1 and TPH1 is computed as will be illustrated in the sample problem.

This is the actual tidal condition case for tidal flux at the ocean-bay opening. Note, this subroutine is <u>not</u> used when the current velocity is specified at this open boundary (i.e. M=1 for specifying $V_o(t)$). It is further pointed out that the user must start the computer run at t=0 (TTOT=0) with a flat surface). ETA(I,J)=0, and ETA=0 at the open boundary. Otherwise, the large step in surface height at the open boundary will result in numerical instability, since the governing equations do <u>not</u> adjust sufficiently fast to yield a compatible and realistic situation between surface heights and currents. Since, the domain of solution is assumed to be "still" (i.e. u=v=Ω=o) initially for the case of not having an adequate initial data base for specifying currents, the above noted specification of ETA(I,J) at t=0 must be followed to insure numerical stability.

```
C***************************************************************************
C      THIS SUBROUTINE CALCULATES THE SURFACE ELEVATION AT THE OCEAN-BAY
C      OPENING AT EVERY TIME STEP
C***************************************************************************
       SUBROUTINE TIDE(IN,JN,TTOT,L,DT,HTD,HI,HTE,INLET,I1,I2,J1,J3,J4
      C,I5,TPH1,A1,A2)
       DIMENSION HTD(IN,JN),HI(IN,JN),HTE(IN,JN)
       ETA=A1+A2*COS((TTOT+TPH1)*(6.28/(12.15*3600.0)))
       IF(INLET.GT.1) GO TO 2000
       DO 40 I=I1,I2
       ITTOT=TTOT
       IF(L.GT.1.OR.ITTOT.GT.0) GO TO 5
       HTD(I,J1)=ETA+HI(I,J1)
       GO TO 40
5      HTE(I,J1)=ETA+HI(I,J1)
40     CONTINUE
       GO TO 1000
2000   CONTINUE
       DO 50 J=J3,J4
       ITTOT=TTOT
       IF(L.GT.1.OR.ITTOT.GT.0) GO TO 15
       HTD(I5,J)=ETA+HI(I5,J)
       GO TO 50
15     HTE(I5,J)=ETA+HI(I5,J)
50     CONTINUE
1000   CONTINUE
       RETURN
       END
```

7.1.5 VEL(M=1)

This subroutine specifies the current velocity at the ocean-bay opening as a function of time (ie TTOT or TTOT + TTOT1) in the form:

$V_o\ (t) = V_o \cos \omega\ (t + t\phi)$ or:

$U_o\ (t) = U_o \cos \omega\ (t + t\phi)$ depending on which axis the open boundary is aligned.

Then, in terms of the program symbols:

Vo = Vo*cos ((TT + TPH) *(6.24)/(12.15 * 3600))

Where Vo is computed as an estimate and TPH is computed as will be illustrated in the sample problem. TT = TTOT or TT = TTOT + TTOT1. Similarly, Vo (t) = 0 at t = 0 is recommended along with an initially 'flat' surface to insure compatibility between the surface heights and the velocity field.

```
C***************************************************************************
C     THIS SUBROUTINE CALCULATES THE CURRENT VELOCITY AT THE OCEAN-BAY
C     OPENING AT EVERY TIME STEP
C***************************************************************************
      SUBROUTINE VEL(IN,JN,KN,U,V,H,G,D,E,TT,VO,TPH,I1,I2,I5,J1,J3,J4,
     CINLET)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN)
     C,D(IN,JN,KN),E(IN,JN,KN)
      KN1=KN-1
      DO 1000 K=1,KN1
      IF(INLET.GT.1) GO TO 2000
      DO 100 I=I1,I2
      V(I,J1,K)=VO*COS((TT+TPH)*(6.28/(12.15*3600.0)))
      G(I,J1,K)=V(I,J1,K)
      E(I,J1,K)=G(I,J1,K)
100   CONTINUE
      GO TO 1000
2000  CONTINUE
      DO 200 J=J3,J4
      U(I5,J,K)=VO*COS((TT+TPH)*(6.28/(12.15*3600.0)))
      H(I5,J,K)=U(I5,J,K)
      D(I5,J,K)=H(I5,J,K)
200   CONTINUE
1000  CONTINUE
      RETURN
      END
```

7.1.6 TIDAL

This subroutine specifies T(I,J,K) as an initial temperature distribution which is constructed by the user from an IR-Data Base. This will be illustrated in the sample problem. Specification of an initial surface as constructed from an adequate tide data base may be inputted into the model but is left optional for the user.

```
C*****************************************************************************
C      THIS SUBROUTINE READS IN THE INITIAL VALUES FOR THE TEMPERATURE
C      DISTRIBUTION AND THE INITIAL VALUES FOR THE SURFACE
C      ELEVATIONS(OPTIONAL)
C*****************************************************************************
       SUBROUTINE TIDAL(IN,JN,KN,ETA,HI,HT,HTD,HTE,II,T,IHITE)
       DIMENSION II(IN),ETA(IN,JN),HI(IN,JN),HT(IN,JN),HTD(IN,JN),
      CHTE(IN,JN),T(IN,JN,KN)
       DO 100 I=1,IN
       READ 5,(T(I,J,1),J=1,JN)
5      FORMAT( )
100    CONTINUE
       DO 101 I=1,IN
       DO 101 J=1,JN
       DO 102 K=2,KN
       T(I,J,K)=T(I,J,1)
102    CONTINUE
101    CONTINUE
       IF(IHITE.EQ.1) GO TO 25
       DO 10 I=1,IN
       READ 1,II(I),(ETA(I,J),J=1,JN)
1      FORMAT(I3,11F7.2)
10     CONTINUE
       DO 20 I=1,IN
       DO 20 J=1,JN
       HT(I,J)=ETA(I,J)+HI(I,J)
       HTD(I,J)=HT(I,J)
       HTE(I,J)=HT(I,J)
20     CONTINUE
25     CONTINUE
       RETURN
       END
```

7.1.7 DATA

This subroutine specifies the wind stresses, equilibrium temperature and the surface heat transfer coefficient every hour. However, these physical parameters must be calculated by the user as will be shown in the sample problem. Note, that this subroutine has been programmed for a maximum computer run of 12 hours. However, this subroutine can be used for less than 12 hours, or it may be easily modified by the user for computer runs in excess of 12 hours, by merely reading in values of these physical parameters for the desired increase in number of hours.

```
C*******************************************************************
C     THIS SUBROUTINE READS IN HOURLY VALUES OF SURFACE WIND SHEAR STRESS,
C     EQUILIBRIUM TEMPERATURE, AND SURFACE HEAT TRANSFER COEFFICIENT
C*******************************************************************
      SUBROUTINE DATA(TAUX,TAUY,TTOT,TA,HS,DTAUX,DTAUY,DTA,DHS,TFLAT)
      DIMENSION DTAUX(13),DTAUY(13),DTA(13),DHS(13)
      TTOT=TTOT-TFLAT
      IF(TTOT.LT.0.0) GO TO 50
      IF(TTOT.GE.0.0.AND.TTOT.LT.3600.0) GO TO 1
      IF(TTOT.LT.7200.AND.TTOT.GE.3600.) GO TO 2
      IF(TTOT.LT.10800.AND.TTOT.GE.7200.) GO TO 3
      IF(TTOT.LT.14400.AND.TTOT.GE.10800.) GO TO 4
      IF(TTOT.LT.18000.AND.TTOT.GE.14400.) GO TO 5
      IF(TTOT.LT.21600.AND.TTOT.GE.18000.) GO TO 6
      IF(TTOT.LT.25200.AND.TTOT.GE.21600.) GO TO 7
      IF(TTOT.LT.28800.AND.TTOT.GE.25200.) GO TO 8
      IF(TTOT.LT.32400.AND.TTOT.GE.28800.) GO TO 9
      IF(TTOT.LT.36000.AND.TTOT.GE.32400.) GO TO 10
      IF(TTOT.LT.39600.AND.TTOT.GE.36000.) GO TO 11
      IF(TTOT.LT.43200.AND.TTOT.GE.39600.) GO TO 12
      IF(TTOT.LT.46800.AND.TTOT.GE.43200.) GO TO 13
1     CONTINUE
      TAUX=DTAUX(1)
      TAUY=DTAUY(1)
      TA=DTA(1)
      HS=DHS(1)
      GO TO 50
2     CONTINUE
      TAUX=DTAUX(2)
      TAUY=DTAUY(2)
      TA=DTA(2)
      HS=DHS(2)
      GO TO 50
3     CONTINUE
      TAUX=DTAUX(3)
      TAUY=DTAUY(3)
      TA=DTA(3)
      HS=DHS(3)
      GO TO 50
4     CONTINUE
      TAUX=DTAUX(4)
      TAUY=DTAUY(4)
      TA=DTA(4)
      HS=DHS(4)
      GO TO 50
5     CONTINUE
      TAUX=DTAUX(5)
      TAUY=DTAUY(5)
      TA=DTA(5)
      HS=DHS(5)
      GO TO 50
6     CONTINUE
      TAUX=DTAUX(6)
      TAUY=DTAUY(6)
      TA=DTA(6)
      HS=DHS(6)
      GO TO 50
    7 CONTINUE
      TAUX=DTAUX(7)
      TAUY=DTAUY(7)
      TA=DTA(7)
      HS=DHS(7)
      GO TO 50
    8 CONTINUE
      TAUX=DTAUX(8)
      TAUY=DTAUY(8)
      TA=DTA(8)
      HS=DHS(8)
      GO TO 50
    9 CONTINUE
      TAUX=DTAUX(9)
      TAUY=DTAUY(9)
      TA=DTA(9)
      HS=DHS(9)
      GO TO 50
   10 CONTINUE
      TAUX=DTAUX(10)
      TAUY=DTAUY(10)
      TA=DTA(10)
      HS=DHS(10)
      GO TO 50
   11 CONTINUE
      TAUX=DTAUX(11)
      TAUY=DTAUY(11)
      TA=DTA(11)
      HS=DHS(11)
      GO TO 50
   12 CONTINUE
      TAUX=DTAUX(12)
      TAUY=DTAUY(12)
      TA=DTA(12)
      HS=DHS(12)
      GO TO 50
   13 CONTINUE
      TAUX=DTAUX(13)
      TAUY=DTAUY(13)
      TA=DTA(13)
      HS=DHS(13)
   50 CONTINUE
      TTOT=TTOT+TFLAT
      RETURN
      END
```

7.1.8 HEIGHT

This subroutine calculates H(x,y) the depth contour with respect to the free surface, at t = Δt (=HTD(I,J)) by forward differencing the surface height equation in time with respect to the initial depth contour matrix H(x,y) at t=0 (=HT(I,J)). Note, this subroutine is used only for the first time cycle. The integration in this subroutine is performed by applying Simpson's Rule.* The general inlet and outlet conditions are specified by reading in from data cards parameters which impose the location of the inlet, either on the upper horizontal boundary or on the left vertical boundary of the grid system. However, this subroutine can be easily modified by the user for having an inlet on the lower horizontal boundary or on tne right vertical boundary. The only change required is respecifying MAR(I,J) corresponding to the inlet location and reading in from data cards the values of I and J which properly locate the inlet.

The derivatives in the integral are obtained by central differencing in space for interior points, including MAR=6 and MAR=8. Three point single sided differencing is performed on the boundaries. These differert schemes are given in Volume I.

* For KN = 5: $$\int_{x1}^{x5} F(x)\,dx \simeq \Delta x \left[1/3F(x_1) + 4/3F(x_2) + 2/3\ F\ (x_3) + 4/3\ F(x_4) + 1/3F\ (x_5) \right]$$

```
C
C*********************************************************************
C      THIS SUBROUTINE CALCULATES THE TOTAL DEPTH AT EACH X-Y LOCATION
C      IN THE DOMAIN FOR THE FIPST TIME STEP USING A FORWARD DIFFERENCING
C      SCHEME IN TIME
C*********************************************************************
       SUBROUTINE HEIGHT(I,J,K,IN,JN,KN,MAR,U,V,HT,HTD,DZ,DT,DX,DY,HDUM,
      CM,I1,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
       DIMENSION MAR(IN,JN),U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),HTD(IN,JN),
      CHDUM(IN,JN)
       KNM1=KN-1
       DO 50 I=1,IN
       DO 50 J=1,JN
       HDUM(I,J)=0.0
       DO 60 K=1,KN
       IF(MAR(I,J).EQ.0) GO TO 50
       IF(MAR(I,J).EQ.11) GO TO 11
       IF(MAR(I,J).EQ.3) GO TO 12
       IF(MAR(I,J).EQ.5) GO TO 19
       IF(MAR(I,J).EQ.2) GO TO 13
       IF(MAR(I,J).EQ.4) GO TO 14
       IF(MAR(I,J).EQ.1) GO TO 20
       IF(MAR(I,J).EQ.7) GO TO 15
       IF(MAR(I,J).EQ.9) GO TO 16
       IF(MAR(I,J).EQ.10) GO TO 17
       IF(MAR(I,J).EQ.6) GO TO 11
       IF(MAR(I,J).EQ.8) GO TO 11
11     D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
       D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
       GO TO 24
12     CONTINUE
       IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 50
       D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
      CU(I+2,J,K))/(2.*DX)
       D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
       GO TO 24
14     CONTINUE
       D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
      CU(I-1,J,K))/(2.*DX)
       D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
       GO TO 24
13     CONTINUE
       D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
       D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
      CV(I,J+2,K))/(2.*DY)
       GO TO 24
20     CONTINUE
       IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 50
       D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
       D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
      CV(I,J-1,K))/(2.*DY)
       GO TO 24
15     CONTINUE
       D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
      CU(I+2,J,K))/(2.*DX)
       D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
   19 CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
   16 CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
   17 CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
   24 CONTINUE
C....SIMPSON'S RULE IS USED FOR INTEGRATION
      IF(K.EQ.1.OR.K.EQ.5) GO TO 101
      IF(K.EQ.2.OR.K.EQ.4) GO TO 102
      HDUM(I,J)=((D1HUX+D1HVY)*DZ*(2./3.))+HDUM(I,J)
      GO TO 103
  101 HDUM(I,J)=((D1HUX+D1HVY)*DZ/3.)+HDUM(I,J)
      GO TO 103
  102 HDUM(I,J)=((D1HUX+D1HVY)*DZ*(4./3.))+HDUM(I,J)
  103 CONTINUE
   60 CONTINUE
      HTD(I,J)=HT(I,J)-HDUM(I,J)
   50 CONTINUE
      RETURN
      END
```

7.1.9 HEILN

This subroutine calculates H at time level n+l (=HTE(I,J)) from H at time level n(=HTD(I,J)) and H at time level n-l(=HT(I,J)) by using central differencing in time.

Volume I gives the detailed finite difference scheme used by this subroutine for solving the surface height equation. The integration, once again, is performed by using Simpson's Rule. The general inlet and outlet conditions are incorporated as was just discussed for subroutine HEIGHT.

```
C
C***********************************************************************
C      THIS SUBROUTINE CALCULATES THE TOTAL DEPTH AT EACH X-Y LOCATION
C      IN THE DOMAIN FOR THE SECOND TIME STEP AND THEREAFTER USING A CENTRAL
C      DIFFERENCING SCHEME IN TIME
C***********************************************************************
      SUBROUTINE HEILN(IN,JN,KN,MAR,U,V,HT,HTD,HTE,DZ,DT,DX,DY,HDUM,M
     C,I1,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
      DIMENSION MAR(IN,JN),U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),HTD(IN,JN),
     CHDUM(IN,JN),HTE(IN,JN)
      KNM1=KN-1
      DO 50 I=1,IN
      DO 50 J=1,JN
      HDUM(I,J)=0.0
      DO 60 K=1,KN
      IF(MAR(I,J).EQ.0) GO TO 50
      IF(MAR(I,J).EQ.11) GO TO 11
      IF(MAR(I,J).EQ.3) GO TO 12
      IF(MAR(I,J).EQ.5) GO TO 19
      IF(MAR(I,J).EQ.2) GO TO 13
      IF(MAR(I,J).EQ.1) GO TO 20
      IF(MAR(I,J).EQ.4) GO TO 14
      IF(MAR(I,J).EQ.7) GO TO 15
      IF(MAR(I,J).EQ.9) GO TO 16
      IF(MAR(I,J).EQ.10) GO TO 17
      IF(MAR(I,J).EQ.6) GO TO 11
      IF(MAR(I,J).EQ.8) GO TO 11
11    D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
      GO TO 24
12    CONTINUE
      IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 50
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
      GO TO 24
14    CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
      GO TO 24
13    CONTINUE
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
20    CONTINUE
      IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 50
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
15    CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
 19   CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
 16   CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
 17   CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
 24   CONTINUE
C....SIMPSON'S RULE IS USED FOR INTEGRATION
      IF(K.EQ.1.OR.K.EQ.5) GO TO 101
      IF(K.EQ.2.OR.K.EQ.4) GO TO 102
     HDUM(I,J)=((D1HUX+D1HVY)*DZ*(2./3.))+HDUM(I,J)
      GO TO 103
 101  HDUM(I,J)=((D1HUX+D1HVY)*DZ/3.)+HDUM(I,J)
      GO TO 103
 102  HDUM(I,J)=((D1HUX+D1HVY)*DZ*(4./3.))+HDUM(I,J)
 103  CONTINUE
   60 CONTINUE
     HTE(I,J)=HTD(I,J)-HDUM(I,J)*2*DT
 50   CONTINUE
      RETURN
      END
```

7.1.10 UVVEL

This subroutine calculates the horizontal components of velocity, u and v, at t = Δt (=H(I,J,K) and G(I,J,K)), respectively) from u and v at t=0 (=U(I,J,K) and V(I,J,K)) by using a forward differencing in time.
Volume I details how the u and v momentum equations are solved. Note, this subroutine is used only for the first time cycle. The general inlet and outlet conditions are specified by reading in from data cards values of the parameters which set the location properly or the boundary. Modification, as mentioned earlier in the description of subroutine HEIGHT, may be easily incorporated by the user.

The spatial derivatives have been replaced by central differencing in the interior of the domain and three point single sided differencing on the boundaries, as shown in Volume I. Again, MAR=6 and MAR=8 boundary corners are tested as interior points.

```
C
C***********************************************************************
C     THIS SUBROUTINE CALCULATES THE HORIZONTAL VELOCITIES,U,V, AT EACH
C     X-Y LOCATION AND DEPTH IN THE DOMAIN FOR THE FIRST TIME STEP USING
C     A FORWARD DIFFERENCING SCHEME IN TIME
C***********************************************************************
      SUBROUTINE UVVEL(IN,JN,KN,U,V,H,G,DX,DY,DZ,W,TAUX,TAUY,
     CDT,HT,HX,HY,ETA,P,MAR,CI,CC,CP,CH,CV,KH,KV,GR,RR,FF,HTD,RO,T,I1
     C,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6,M)
      REAL KH,KV
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),
     CHT(IN,JN),HX(IN,JN),HY(IN,JN),ETA(IN,JN),
     CP(IN,JN,KN),MAR(IN,JN),HTD(IN,JN),W(IN,JN,KN)
     C,RO(IN,JN,KN),T(IN,JN,KN)
      KN1=KN-1
      DO 10 I=1,IN
      DO 10 J=1,JN
      IF(MAR(I,J).EQ.0) GO TO 10
      IF(MAR(I,J).EQ.5) GO TO 10
      IF(MAR(I,J).EQ.7) GO TO 10
      IF(MAR(I,J).EQ.9) GO TO 10
      IF(MAR(I,J).EQ.10) GO TO 10
      IF(MAR(I,J).EQ.3) GO TO 10
      IF(MAR(I,J).EQ.4) GO TO 10
      DO 8 K=1,KN1
      IF(MAR(I,J).EQ.6) GO TO 11
      IF(MAR(I,J).EQ.8) GO TO 11
      IF(MAR(I,J).EQ.11) GO TO 11
      IF(MAR(I,J).EQ.1) GO TO 101
      IF(MAR(I,J).EQ.2) GO TO 102
11    CONTINUE
      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      D1PX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
      D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)*U(I-1,J,K)*
     CHT(I-1,J))/(2*DX)
      D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-U(I,J-1,K)*V(I,J-1,K)*
     CHT(I,J-1))/(2*DY)
      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
      D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
      D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
      D2UY=(U(I,J+1,K)+U(I,J-1,K)-2*U(I,J,K))/(DY*DY)
      GO TO 100
101   CONTINUE
      IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2) GO TO 10
      DHY=(3*HT(I,J)+HT(I,J-2)-4*HT(I,J-1))/(2*DY)
      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      ETAY=(3*ETA(I,J)+ETA(I,J-2)-4*ETA(I,J-1))/(2*DY)
      D1PX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
      D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)*U(I-1,J,K)*
     CHT(I-1,J))/(2*DX)
      D1HUVY=(3*U(I,J,K)*V(I,J,K)*HT(I,J)+U(I,J-2,K)*V(I,J-2,K)
     C*HT(I,J-2)-4*U(I,J-1,K)*V(I,J-1,K)*HT(I,J-1))/(2*DY)
      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
      D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
      D1UY=0.0
      D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
      GO TO 100
102   CONTINUE
      IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 10
      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2*DY)
      DHY=(4*HT(I,J+1)-3*HT(I,J)-HT(I,J+2))/(2*DY)
      D1PX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
      D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)*U(I-1,J,K)*
     CHT(I-1,J))/(2*DX)
      D1HUVY=(4*U(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-3*U(I,J,K)*V(I,J,K)
     C*HT(I,J)-U(I,J+2,K)*V(I,J+2,K)*HT(I,J+2))/(2*DY)
      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
      D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
      D1UY=0.0
      D2UY=(U(I,J,K)+U(I,J+2,K)-2*U(I,J+1,K))/(DY*DY)
100   CONTINUE
      RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
      RR=RO(I,J,K)
      IF(K.EQ.1) GO TO 70
      D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
      D2UZ=(U(I,J,K+1)+U(I,J,K-1)-2*U(I,J,K))/(DZ*DZ)
      GO TO 80
70    D1UWZ=(4*U(I,J,K+1)*W(I,J,K+1)-3*U(I,J,K)*W(I,J,K)-U(I,J,K+2)*
     CW(I,J,K+2))/(2.*DZ)
      D2UZ=(2*U(I,J,K+1)+(TAUX/KV)*HT(I,J)*2*DZ-2*U(I,J,K))/(DZ*DZ)
80    CONTINUE
      UI=CI*(D1HUUX+D1HUVY+HT(I,J)*D1UWZ)
      UP=CP*HT(I,J)*(ETAX*GR)*(-1.)
      UC=CC*HT(I,J)*FF*V(I,J,K)
      UH=CH*KH*(HX(I,J)*D1UX+ETAX*D1UX+HT(I,J)*D2UX)+CH*KH*(HY(I,J)*D1UY
     C+ETAY*D1UY+HT(I,J)*D2UY)
      UV=CV*KV*D2UZ/HT(I,J)
      H(I,J,K)= ((-UI+UP-UC+UH+UV)*DT+HT(I,J)*U(I,J,K))/HTD(I,J)
8     CONTINUE
10    CONTINUE
      IF(M.EQ.1) GO TO 7000
      IF(INLET.EQ.1) GO TO 7000
      DO 6000 K=1,KN1
      SUM=0.0
      DO 6002 J=J3,J4
      SUM=SUM+H(I5+1,J,K)
6002  CONTINUE
      DO 6003 J=J3,J4
      H(I5+1,J,K)=SUM/(J4-J3+1)
6003  CONTINUE
6000  CONTINUE
7000  CONTINUE
      DO 970 I=1,IN
      DO 970 J=1,JN
      DO 960 K=1,KN1
      IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 5002
      IF(I.EQ.I6.AND.J.GE.J5.AND.J.LE.J6) GO TO 2005
      GO TO 970
5002  H(I,J,K)=H(I+1,J,K)
      GO TO 970
2005  H(I,J,K)=H(I-1,J,K)
```

```
  960 CONTINUE
  970 CONTINUE
      DO 30 I=1,IN
      DO 30 J=1,JN
      IF(MAR(I,J).EQ.0) GO TO 30
      IF(MAR(I,J).EQ.5) GO TO 30
      IF(MAR(I,J).EQ.7) GO TO 30
      IF(MAR(I,J).EQ.9) GO TO 30
      IF(MAR(I,J).EQ.10) GO TO 30
      DO 7 K=1,KN1
      IF(MAR(I,J).EQ.6) GO TO 12
      IF(MAR(I,J).EQ.8) GO TO 12
      IF(MAR(I,J).EQ.11) GO TO 12
      IF(MAR(I,J).EQ.1) GO TO 201
      IF(MAR(I,J).EQ.2) GO TO 202
      IF(MAR(I,J).EQ.3) GO TO 203
      IF(MAR(I,J).EQ.4) GO TO 204
   12 CONTINUE
      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      D1PY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
      D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)*V(I-1,J,K)*
     CHT(I-1,J))/(2*DX)
      D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-V(I,J-1,K)*V(I,J-1,K)*
     CHT(I,J-1))/(2*DY)
      D1VX=(V(I+1,J,K)-V(1-1,J,K))/(2*DX)
      D2VX=(V(I+1,J,K)+V(I-1,J,K)-2*V(I,J,K))/(DX*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
      GO TO 200
  201 CONTINUE
      GO TO 30
  202 CONTINUE
      GO TO 30
  203 CONTINUE
      IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4) GO TO 30
      ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2*DX)
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      DHX=(4*HT(I+1,J)-3*HT(I,J)-HT(I+2,J))/(2*DX)
      D1PY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
      D1HUVX=(4*U(I+1,J,K)*V(I+1,J,K)*HT(I+1,J)-3*U(I,J,K)*V(I,J,K)
     C*HT(I,J)-U(I+2,J,K)*V(I+2,J,K)*HT(I+2,J))/(2*DX)
      D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-V(I,J-1,K)*V(I,J-1,K)*
     CHT(I,J-1))/(2*DY)
      D1VX=0.0
      D2VX=(V(I,J,K)+V(I+2,J,K)-2*V(I+1,J,K))/(DX*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
      GO TO 200
  204 CONTINUE
      IF(I.EQ.I6.AND.J.GE.J5.AND.J.LE.J6) GO TO 30
      ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2*DX)
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      DHX=(3*HT(I,J)+HT(I-2,J)-4*HT(I-1,J))/(2*DX)
      D1PY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
      D1HUVX=(3*U(I,J,K)*V(I,J,K)*HT(I,J)+U(I-2,J,K)*V(I-2,J,K)
     C*HT(I-2,J)-4*U(I-1,J,K)*V(I-1,J,K)*HT(I-1,J))/(2*DX)
```

```
      D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-V(I,J-1,K)*V(I,J-1,K)*
     CHT(I,J-1))/(2*DY)
      D1VX=0.0
      D2VX=(V(I,J,K)+V(I-2,J,K)-2*V(I-1,J,K))/(DX*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
  200 CONTINUE
      RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
      RR=RO(I,J,K)
      IF(K.EQ.1) GO TO 90
      D1VWZ=(V(I,J,K+1)*W(I,J,K+1)-V(I,J,K-1)*W(I,J,K-1))/(2*DZ)
      D2VZ=(V(I,J,K+1)+V(I,J,K-1)-2*V(I,J,K))/(DZ*DZ)
      GO TO 95
   90 D1VWZ=(4*V(I,J,K+1)*W(I,J,K+1)-3*V(I,J,K)*W(I,J,K)-V(I,J,K+2)*
     CW(I,J,K+2))/(2.*DZ)
      D2VZ=(2*V(I,J,K+1)+(TAUY/KV)*HT(I,J)*2*DZ-2*V(I,J,K))/(DZ*DZ)
   95 CONTINUE
   60 VI=CI*(D1HUVX+D1HVVY+HT(I,J)*D1VWZ)
      VP=CP*HT(I,J)*(ETAY*GR)*(-1.)
      VC=CC*HT(I,J)*FF*U(I,J,K)
      VH=CH*KH*(HX(I,J)*D1VX+ETAX*D1VX+HT(I,J)*D2VX)+CH*KH*(HY(I,J)*D1VY
     C+ETAY*D1VY+HT(I,J)*D2VY)
      VV=CV*KV*D2VZ/HT(I,J)
      G(I,J,K)= ((-VI+VP+VC+VH+VV)*DT+HT(I,J)*V(I,J,K))/HT3(I,J)
    7 CONTINUE
   30 CONTINUE
      IF(M.EQ.1) GO TO 700
      IF(INLET.GT.1) GO TO 700
      DO 600 K=1,KN1
      SUM=0.0
      DO 602 I=I1,I2
      SUM=SUM+G(I,J1-1,K)
  602 CONTINUE
      DO 603 I=I1,I2
      G(I,J1-1,K)=SUM/(I2-I1+1)
  603 CONTINUE
  600 CONTINUE
  700 CONTINUE
      DO 97 I=1,IN
      DO 97 J=1,JN
      DO 96 K=1,KN1
      IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 205
      IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 502
      GO TO 97
  205 CONTINUE
      G(I,J,K)=G(I,J+1,K)
      GO TO 97
  502 CONTINUE
      G(I,J,K)=G(I,J-1,K)
   96 CONTINUE
   97 CONTINUE
      RETURN
      END
```

7.1.11 UVVELN

This subroutine calculates the horizontal components of velocity u and v at time level n+1 (=D(I,J,K) and E(I,J,K) respectively) from u and v at time level n (=H(I,J,K) and G(I,J,K) and u and v at time level n-1 (=U(I,J,K) and V(I,J,K)) by using central differencing in time. The numerical scheme used for solving these equations is given in Volume I. Again, the general inlet and outlet conditions are specified and may be modified by the user. Note, DuFort-Frankel differencing is used for the vertical momentum diffusion term. The spatial derivatives have been differenced as shown in Volume I.

```
C
C***********************************************************************
C      THIS SUBROUTINE CALCULATES THE HORIZONTAL VELOCITIES, U,V, AT EACH
C      X-Y LOCATION AND DEPTH IN THE DOMAIN FOR THE SECOND TIME STEP AND
C      THEREAFTER USING A CENTRAL DIFFERENCING SCHEME IN TIME
C***********************************************************************
      SUBROUTINE UVVELN(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
     CHT,HTD,HTE,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,CP,CC,CI,CH,CV,RO,T,I1
     C,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6,M)
      REAL KH,KV
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),
     CD(IN,JN,KN),E(IN,JN,KN),HT(IN,JN),HTD(IN,JN),HTE(IN,JN),HX(IN,JN),
     CHY(IN,JN),ETA(IN,JN),P(IN,JN,KN),MAR(IN,JN),W(IN,JN,KN)
     C,RO(IN,JN,KN),T(IN,JN,KN)
      KN1=KN-1
      DO 10 I=1,IN
      DO 10 J=1,JN
      IF(MAR(I,J).EQ.0) GO TO 10
      IF(MAR(I,J).EQ.5) GO TO 10
      IF(MAR(I,J).EQ.7) GO TO 10
      IF(MAR(I,J).EQ.9) GO TO 10
      IF(MAR(I,J).EQ.10) GO TO 10
      IF(MAR(I,J).EQ.3) GO TO 10
      IF(MAR(I,J).EQ.4) GO TO 10
      DO 8 K=1,KN1
      IF(MAR(I,J).EQ.6) GO TO 11
      IF(MAR(I,J).EQ.8) GO TO 11
      IF(MAR(I,J).EQ.11) GO TO 11
      IF(MAR(I,J).EQ.1) GO TO 101
      IF(MAR(I,J).EQ.2) GO TO 102
11    CONTINUE
      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
      D1PX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
      D1HUUX=(H(I+1,J,K)*H(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*
     CH(I-1,J,K)*HTD(I-1,J))/(2*DX)
      D1HUVY=(H(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-H(I,J-1,K)*
     CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
      D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
      D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
      D2UY=(U(I,J+1,K)+U(I,J-1,K)-2*U(I,J,K))/(DY*DY)
      GO TO 100
101   CONTINUE
      IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2) GO TO 10
      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
      DHY=(3*HT(I,J)+HT(I,J-2)-4*HT(I,J-1))/(2*DY)
      D1PX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
      D1HUUX=(H(I+1,J,K)*H(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*
     CH(I-1,J,K)*HTD(I-1,J))/(2*DX)
      D1HUVY=(3*H(I,J,K)*G(I,J,K)*HTD(I,J)+H(I,J-2,K)*G(I,J-2,K)
     C*HTD(I,J-2)-4*H(I,J-1,K)*G(I,J-1,K)*HTD(I,J-1))/(2*DY)
      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
      D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
```

```
        D1UY=0.0
        D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
        GO TO 100
102     CONTINUE
        IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 10
        ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
        DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
        DHY=(4*HT(I,J+1)-3*HT(I,J)-HT(I,J+2))/(2*DY)
        D1PX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
        D1HUUX=(H(I+1,J,K)*H(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*
       CH(I-1,J,K)*HTD(I-1,J))/(2*DX)
        D1HUVY=(4*H(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-3*H(I,J,K)*G(I,J,K)
       C*HTD(I,J)-H(I,J+2,K)*G(I,J+2,K)*HTD(I,J+2))/(2*DY)
        D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
        D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
        D1UY=0.0
        D2UY=(U(I,J,K)+U(I,J+2,K)-2*U(I,J+1,K))/(DY*DY)
100     CONTINUE
        RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
        RR=RO(I,J,K)
        IF(K.EQ.1) GO TO 70
        D1UWZ=(H(I,J,K+1)*W(I,J,K+1)-H(I,J,K-1)*W(I,J,K-1))/(2*DZ)
        UZ=H(I,J,K+1)+H(I,J,K-1)-U(I,J,K)
        GO TO 80
70      D1UWZ=(4*H(I,J,K+1)*W(I,J,K+1)-3*H(I,J,K)*W(I,J,K)-H(I,J,K+2)*
       CW(I,J,K+2))/(2.*DZ)
        UZ=2*H(I,J,K+1)+(TAUX/KV)*HTD(I,J)*2*DZ-U(I,J,K)
8C      CONTINUE
        UI=CI*(D1HUUX+D1HUVY+HTD(I,J)*D1UWZ)
        UP=CP*HTD(I,J)*(ETAX*GR)*(-1.)
        UC=CC*HTD(I,J)*FF*G(I,J,K)
        UH=CH*KH*(DHX*D1UX+HT(I,J)*D2UX)+CH*KH*(DHY*D1UY+HT(I,J)*D2UY)
        D(I,J,K)=(((-UI+UP-UC+UH)*2*DT+((CV*KV*UZ)/(DZ*DZ*HTD(I,J)))*2*
       CDT+U(I,J,K)*HT(I,J))/HTE(I,J))/(1.+(CV*KV/(DZ*DZ*HTD(I,J)))*2*DT
       C/HTE(I,J))
8       CONTINUE
10      CONTINUE
        IF(M.EQ.1) GO TO 7000
        IF(INLET.EQ.1) CO TO 7000
        DO 6000 K=1,KN1
        SUM=0.0
        DO 6002 J=J3,J4
        SUM=SUM+D(I5+1,J,K)
6002    CONTINUE
        DO 6003 J=J3,J4
        D(I5+1,J,K)=SUM/(J4-J3+1)
6003    CONTINUE
6000    CONTINUE
7000    CONTINUE
        DO 970 I=1,IN
        DO 970 J=1,JN
        DO 960 K=1,KN1
        IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 5002
        IF(I.EQ.I6.AND.J.GE.J5.AND.J.LE.J6) GO TO 2005
        GO TO 970
5002    D(I,J,K)=D(I+1,J,K)
        GO TO 970
```

```
 2005 D(I,J,K)=D(I-1,J,K)
 960  CONTINUE
 970  CONTINUE
      DO 30 I=1,IN
      DO 30 J=1,JN
      IF(MAR(I,J).EQ.0) GO TO 30
      IF(MAR(I,J).EQ.5) GO TO 30
      IF(MAR(I,J).EQ.7) GO TO 30
      IF(MAR(I,J).EQ.9) GO TO 30
      IF(MAR(I,J).EQ.10) GO TO 30
      DO 7 K=1,KN1
      IF(MAR(I,J).EQ.6) GO TO 12
      IF(MAR(I,J).EQ.8) GO TO 12
      IF(MAR(I,J).EQ.11) GO TO 12
      IF(MAR(I,J).EQ.1) GO TO 201
      IF(MAR(I,J).EQ.2) GO TO 202
      IF(MAR(I,J).EQ.3) GO TO 203
      IF(MAR(I,J).EQ.4) GO TO 204
 12   CONTINUE
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
      D1PY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
      D1HUVX=(H(I+1,J,K)*G(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*G(I-1,J,K)*
     CHTD(I-1,J))/(2*DX)
      D1HVVY=(G(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-G(I,J-1,K)*
     CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
      D1VX=(V(I+1,J,K) V(I-1,J,K))/(2*DX)
      D2VX=(V(I+1,J,K)+V(I-1,J,K)-2*V(I,J,K))/(DX*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
      GO TO 200
 201  CONTINUE
      GO TO 30
 202  CONTINUE
      GO TO 30
 203  CONTINUE
      IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4) GO TO 30
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      DHX=(4*HT(I+1,J)-3*HT(I,J)-HT(I+2,J))/(2*DX)
      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
      D1PY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
      D1HUVX=(4*H(I+1,J,K)*G(I+1,J,K)*HTD(I+1,J)-3*H(I,J,K)*G(I,J,K)
     C*HTD(I,J)-H(I+2,J,K)*G(I+2,J,K)*HTD(I+2,J))/(2*DX)
      D1HVVY=(G(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-G(I,J-1,K)*
     CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
      D1VX=0.0
      D2VX=(V(I,J,K)+V(I+2,J,K)-2*V(I+1,J,K))/(DX*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
      GO TO 200
 204  CONTINUE
      IF(I.EQ.I6.AND.J.GE.J5.AND.J.LE.J6) GO TO 30
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      DHX=(3*HT(I,J)+HT(I-2,J)-4*HT(I-1,J))/(2*DX)
      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
      D1PY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
```

```
       D1HUVX=(3*H(I,J,K)*G(I,J,K)*HTD(I,J)+H(I-2,J,K)*G(I-2,J,K)
      C*HTD(I-2,J)-4*H(I-1,J,K)*G(I-1,J,K)*HTD(I-1,J))/(2*DX)
       D1HVVY=(G(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-G(I,J-1,K)*
      CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
       D1VX=0.0
       D2VX=(V(I,J,K)+V(I-2,J,K)-2*V(I-1,J,K))/(DX*DX)
       D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
       D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
 200   CONTINUE
       RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
       RR=RO(I,J,K)
       IF(K.EQ.1) GO TO 90
       D1VWZ=(G(I,J,K+1)*W(I,J,K+1)-G(I,J,K-1)*W(I,J,K-1))/(2*DZ)
       VZ=G(I,J,K+1)+G(I,J,K-1)-V(I,J,K)
       GO TO 95
 90    D1VWZ=(4*G(I,J,K+1)*W(I,J,K+1)-3*G(I,J,K)*W(I,J,K)-G(I,J,K+2)*
      CW(I,J,K+2))/(2.*DZ)
       VZ=2*G(I,J,K+1)+(TAUY/KV)*HTD(I,J)*2*DZ-V(I,J,K)
 95    CONTINUE
       VI=CI*(D1HUVX+D1HVVY+HTD(I,J)*D1VWZ)
       VP=CP*HTD(I,J)*(ETAY*GR)*(-1.)
       VC=CC*HTD(I,J)*FF*H(I,J,K)
       VH=CH*KH*(DHX*D1VX+HT(I,J)*D2VX)+CH*KH*(DHY*D1VY+HT(I,J)*D2VY)
       E(I,J,K)=(((-VI+VP+VC+VH)*2*DT+((CV*KV*VZ)/(DZ*DZ*HTJ(I,J)))*2*DT
      C+V(I,J,K)*HT(I,J))/HTE(I,J))/(1.+(CV*KV/(DZ*DZ*HTD(I,J)))*2*DT
      C/HTE(I,J))
 7     CONTINUE
 30    CONTINUE
       IF(M.EQ.1) GO TO 700
       IF(INLET.GT.1) GO TO 700
       DO 600 K=1,KN1
       SUM=0.0
       DO 602 I=I1,I2
       SUM=SUM+E(I,J1-1,K)
 602   CONTINUE
       DO 603 I=I1,I2
       E(I,J1-1,K)=SUM/(I2-I1+1)
 603   CONTINUE
 600   CONTINUE
 700   CONTINUE
       DO 97 I=1,IN
       DO 97 J=1,JN
       DO 96 K=1,KN1
       IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 205
       IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 502
       GO TO 97
 205   CONTINUE
       E(I,J,K)=E(I,J+1,K)
       GO TO 97
 502   CONTINUE
       E(I,J,K)=E(I,J-1,K)
 96    CONTINUE
 97    CONTINUE
       RETURN
       END
```

7.1.12 WVEL

This subroutine calculates the equivalent vertical velocity Ω(I.J,K) in the α, β, σ (=x,y,σ) coordinate system at each x, y location and depth σ in the domain Ω(I,J,K) at t = Δt (=W(I,J,K)) is calculated from u, v, and H at t = Δt (=H(I,J,K)) G(I,J,K), HTD(I,J,K)) as shown in Volume I. Thereafter, Ω(I,J,K) at time level n+1 is calculated from u,v and H at time level n+1 (=D(I,J,K), E(I,J,K), HTE(I,J)) Simpson's Rule is used for performing the integration.

```
C
C************************************************************************
C      THIS SUBROUTINE CALCULATES THE EQUIVALENT VERTICAL VELOCITY IN THE SI
C      COORDINATE SYSTEM AT EACH X-Y LOCATION AND DEPTH IN THE DOMAIN
C      AT EACH TIME STEP
C************************************************************************
       SUBROUTINE WVEL(IN,JN,KN,U,V,W,HT,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
       DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),W(IN,JN,KN),MAR(IN,JN)
       KN1=KN-1
       DO 10 I=1,IN
       DO 10 J=1,JN
       DUM=0.
       DO 9 K=1,KN
       IF(MAR(I,J).EQ.0) GO TO 10
       IF(MAR(I,J).EQ.11) GO TO 11
       IF(MAR(I,J).EQ.6) GO TO 11
       IF(MAR(I,J).EQ.8) GO TO 11
       IF(MAR(I,J).EQ.1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 12
       IF(MAR(I,J).EQ.3.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 12
       IF(MAR(I,J).LT.11) GO TO 10
11     D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
       D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
       GO TO 24
12     CONTINUE
       CALL WVEL1(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,D1HUX,D1HVY)
24     CONTINUE
       IF(K.EQ.1.OR.K.EQ.5) GO TO 27
       IF(K.EQ.2.OR.K.EQ.4) GO TO 28
       DUM=DUM+DZ*(2./3.)*(D1HUX+D1HVY)/HT(I,J)
       GO TO 9
27     DUM=DUM+(DZ/3.)*(D1HUX+D1HVY)/HT(I,J)
       GO TO 9
28     DUM=DUM+DZ*(4./3.)*(D1HUX+D1HVY)/HT(I,J)
9      CONTINUE
       WUD=0.
       DO 8 K=2,KN
       IF(MAR(I,J).EQ.0) GO TO 10
       IF(MAR(I,J).EQ.11) GO TO 111
       IF(MAR(I,J).EQ.6) GO TO 111
       IF(MAR(I,J).EQ.8) GO TO 111
       IF(MAR(I,J).EQ.1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 112
       IF(MAR(I,J).EQ.3.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 112
       IF(MAR(I,J).LT.11) GO TO 10
111    D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
       D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
       D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
       D1HVY1=(HT(I,J+1)*V(I,J+1,K-1)-HT(I,J-1)*V(I,J-1,K-1))/(2.*DY)
       GO TO 200
112    CONTINUE
       CALL WVEL2(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,D1HUX,D1HVY,
      CD1HUX1,D1HVY1)
200    CONTINUE
       IF(K.EQ.2) GO TO 101
       IF(K.EQ.4) GO TO 101
       IF(K.EQ.3) GO TO 102
102    WUD=WUD+DZ*((2./3.)*(D1HUX1+D1HVY1)+(1./3.)*(D1HUX+D1HVY))/
```

```
      CHT(I,J)
       GO TO 300
 101   WUD=WUD+DZ*((1./3.)*(D1HUX1+D1HVY1)+(2./3.)*(D1HUX+D1HVY))/
      CHT(I,J)
       GO TO 300
 300   CONTINUE
       W(I,J,K)=-WUD+DUM*(K-1)*DZ
 8     CONTINUE
 10    CONTINUE
       RETURN
       END
```

7.1.13 WVEL1

This subroutine calculates the differential terms, in diff-erenced form, in the definite integral for the equivalent vertical velocity, at each time step, from u, v, H at $t=\Delta t$, and at the time level n+1, thereafter.

```
C*****************************************************************************
C     THIS SUBROUTINE CALCULATES THE DIFFERENTIAL(DIFFERENCED) TERMS IN THE
C     DEFINITE INTEGRAL FOR THE EQUIVALENT VERTICAL VELOCITY AT
C     EACH TIME STEP
C*****************************************************************************
      SUBROUTINE WVEL1(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,D1HUX,D1HVY)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),MAR(IN,JN)
      IF(MAR(I,J).EQ.3) GO TO 12
      IF(MAR(I,J).EQ.5) GO TO 19
      IF(MAR(I,J).EQ.2) GO TO 13
      IF(MAR(I,J).EQ.1) GO TO 20
      IF(MAR(I,J).EQ.4) GO TO 14
      IF(MAR(I,J).EQ.7) GO TO 15
      IF(MAR(I,J).EQ.9) GO TO 16
      IF(MAR(I,J).EQ.10) GO TO 17
12    CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
      GO TO 24
14    CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
      GO TO 24
13    CONTINUE
      IF(J.EQ.1.AND.I.GE.31.AND.I.LE.33) GO TO 31
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
31    CONTINUE
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
20    CONTINUE
      IF(J.EQ.11.AND.I.GE.7.AND.I.LE.16) GO TO 32
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
32    CONTINUE
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
15    CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
19    CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
```

```
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      GO TO 24
 16   CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I.J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      GO TO 24
 17   CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
 24   CONTINUE
      RETURN
      END
```

7.1.14 WVEL2

This subroutine calculates the differential time, in differenced form, in the indefinite integral for the equivalent vertical velocitv at each time step. u, v, and HT are used at t = Δt and D, E, HTE are used at time level n+1, thereafter.

```
C**********************************************************************
C     THIS SUBROUTINE CALCULATES THE DIFFERENTIAL(DIFFERENCED) TERMS IN
C     THE INDEFINITE INTEGRAL FOR THE EQUIVALENT VERTICAL VELOCITY
C     AT EACH TIME STEP
C**********************************************************************
      SUBROUTINE WVEL2(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,D1HUX,D1HVY,
     CD1HUX1,D1HVY1)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),MAR(IN,JN)
      IF(MAR(I,J).EQ.3) GO TO 112
      IF(MAR(I,J).EQ.5) GO TO 119
      IF(MAR(I,J).EQ.2) GO TO 113
      IF(MAR(I,J).EQ.1) GO TO 120
      IF(MAR(I,J).EQ.4) GO TO 114
      IF(MAR(I,J).EQ.7) GO TO 115
      IF(MAR(I,J).EQ.9) GO TO 116
      IF(MAR(I,J).EQ.10) GO TO 117
112   CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HUX1=(4*HT(I+1,J)*U(I+1,J,K-1)-3*HT(I,J)*U(I,J,K-1)-HT(I+2,J)*
     CU(I+2,J,K-1))/(2.*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
      D1HVY1=(HT(I,J+1)*V(I,J+1,K-1)-HT(I,J-1)*V(I,J-1,K-1))/(2.*DY)
      GO TO 200
114   CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HUX1=(3*HT(I,J)*U(I,J,K-1)+HT(I-2,J)*U(I-2,J,K-1)-4*HT(I-1,J)*
     CU(I-1,J,K-1))/(2.*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
      D1HVY1=(HT(I,J+1)*V(I,J+1,K-1)-HT(I,J-1)*V(I,J-1,K-1))/(2.*DY)
      GO TO 200
113   CONTINUE
      IF(J.EQ.1.AND.I.GE.31.AND.I.LE.33) GO TO 311
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      D1HVY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
     CV(I,J+2,K-1))/(2.*DY)
      GO TO 200
311   CONTINUE
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      D1HVY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
     CV(I,J+2,K-1))/(2.*DY)
      GO TO 200
120   CONTINUE
      IF(J.EQ.1).AND.I.GE.7.AND.I.LE.16) GO TO 321
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      D1HVY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*
     CV(I,J-1,K-1))/(2.*DY)
      GO TO 200
  321 CONTINUE
      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      D1HVY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*
     CV(I,J-1,K-1))/(2.*DY)
      GO TO 200
  115 CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HUX1=(4*HT(I+1,J)*U(I+1,J,K-1)-3*HT(I,J)*U(I,J,K-1)-HT(I+2,J)*
     CU(I+2,J,K-1))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      D1HVY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
     CV(I,J+2,K-1))/(2.*DY)
      GO TO 200
  119 CONTINUE
      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
     CU(I+2,J,K))/(2.*DX)
      D1HUX1=(4*HT(I+1,J)*U(I+1,J,K-1)-3*HT(I,J)*U(I,J,K-1)-HT(I+2,J)*
     CU(I+2,J,K-1))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      D1HVY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*
     CV(I,J-1,K-1))/(2.*DY)
      GO TO 200
  116 CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HUX1=(3*HT(I,J)*U(I,J,K-1)+HT(I-2,J)*U(I-2,J,K-1)-4*HT(I-1,J)*
     CU(I-1,J,K-1))/(2.*DX)
      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
     CV(I,J+2,K))/(2.*DY)
      D1HVY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
     CV(I,J+2,K-1))/(2.*DY)
      GO TO 200
  117 CONTINUE
      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
     CU(I-1,J,K))/(2.*DX)
      D1HUX1=(3*HT(I,J)*U(I,J,K-1)+HT(I-2,J)*U(I-2,J,K-1)-4*HT(I-1,J)*
     CU(I-1,J,K-1))/(2.*DX)
      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
     CV(I,J-1,K))/(2.*DY)
      D1HVY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*
     CV(I,J-1,K-1))/(2.*DY)
  200 CONTINUE
      RETURN
      END
```

7.1.15 WW

This subroutine converts Ω to W, that is the vertical component of velocity. The following analytic expression is used for this conversion:

$$W = \Omega H + (\sigma)\frac{dh}{dt} + (\sigma - 1)\ \frac{d\eta}{dt}$$

The actual vertical velocity component, W is defined as WZ(I,J,K) in the model program, and it is calculated at each x, y,σ. Since WZ(I,J,K) is <u>not</u> used in solving the system of governing equations, this subroutine is used only after the last time cycle for each computer run.

```
C*****************************************************************************
C      THIS SUBROUTINE TRANSFORMS THE EQUIVALENT VERTICAL VELOCITY
C      (IN THE SIGMA COORDINATE SYSTEM) INTO THE ACTUAL VERTICAL VELOCITY
C      (IN THE X-Y-Z COORDINATE SYSTEM) AT EACH X-Y LOCATION AND DEPTH
C      IN THE DOMAIN
C*****************************************************************************
       SUBROUTINE WW(IN,JN,KN,HTD,HTE,ETA,D,E,W,WZ,MAR,DX,DY,DZ,DT,HX,HY)
       DIMENSION HTD(IN,JN),HTE(IN,JN),ETA(IN,JN),W(IN,JN,KN),
      CWZ(IN,JN,KN),MAR(IN,JN),D(IN,JN,KN),E(IN,JN,KN),HX(IN,JN)
      C,HY(IN,JN)
       KN1=KN-1
       DO 10 I=1,IN
       DO 10 J=1,JN
       DO 9 K=1,KN1
       WZ(I,J,K)=0.0
       IF(MAR(I,J).EQ.0) GO TO 9
       IF(MAR(I,J).EQ.11) GO TO 11
       IF(MAR(I,J).EQ.6) GO TO 11
       IF(MAR(I,J).EQ.8) GO TO 11
       IF(MAR(I,J).EQ.1) GO TO 101
       IF(MAR(I,J).EQ.2) GO TO 102
       IF(MAR(I,J).EQ.3) GO TO 103
       IF(MAR(I,J).EQ.4) GO TO 104
       IF(MAR(I,J).EQ.5) GO TO 105
       IF(MAR(I,J).EQ.7) GO TO 107
       IF(MAR(I,J).EQ.9) GO TO 109
       IF(MAR(I,J).EQ.10) GO TO 110
 11    CONTINUE
       ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2.*DX)
       ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2.*DY)
       GO TO 100
 101   ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2.*DX)
       ETAY=(3*ETA(I,J)+ETA(I,J-2)-4*ETA(I,J-1))/(2.*DY)
       GO TO 100
 102   ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2.*DX)
       ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2.*DY)
       GO TO 100
 103   ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2.*DX)
       ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2.*DY)
       GO TO 100
 104   ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2.*DX)
       ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2.*DY)
       GO TO 100
 105   ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2.*DX)
       ETAY=(3*ETA(I,J)+ETA(I,J-2)-4*ETA(I,J-1))/(2.*DY)
       GO TO 100
 107   ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2.*DX)
       ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2.*DY)
       GO TO 100
 109   ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2.*DX)
       ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2.*DY)
       GO TO 100
 110   ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2.*DX)
       ETAY=(3*ETA(I,J)+ETA(I,J-2)-4*ETA(I,J-1))/(2.*DY)
 100   CONTINUE
       WZ(I,J,K)=HTE(I,J)*W(I,J,K)+((K-1)*DZ-1.)*((HTE(I,J)-HTD(I,J))/DT
     C+D(I,J,K)*ETAX+E(I,J,K)*ETAY)+((K-1)*DZ)*(D(I,J,K)
     C*HX(I,J)+E(I,J,K)*HY(I,J))
    9 CONTINUE
   10 CONTINUE
      RETURN
      END
```

7.1.16 PRES

This subroutine calculates the pressure field at time level n+1 by knowing the contour depth, H and density, ρ at n+1. Note, that this is the integrated form of the hydrostatic equation. The integration is performed by applying tha trapezoidal rule.

```
C
C***********************************************************************
C     THIS SUBROUTINE CALCULATES THE PRESSURE FIELD
C***********************************************************************
      SUBRCUTINE PRES(IN,JN,KN,HT,RO,GR,P,DZ)
      DIMENSION HT(IN,JN),RO(IN,JN,KN),P(IN,JN,KN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      P(I,J,1)=0.0
      DO 8 K=2,KN
      P(I,J,K)=P(I,J,K-1)+GR*HT(I,J)*(RO(I,J,K-1)+RO(I,J,K))*DZ/2.0
 8    CONTINUE
 10   CONTINUE
      RETURN
      END
```

7.1.17 TEMP

This subroutine calculater the temperature distribution T(I,J,K) at x, y,σ at each time step using a forward differencing in time. T(x, y, σ) at t= Δt (= T(I,J,K) is calculated from T(x,y,σ) at t=0 (=TN1(I,J,K)). Thereafter, T at time level n+1 (=TN1(I,J,K)) is calculated from T at time level n(=T(I,J,K)). The spatial derivatives, once again, have been approximated by central differencing in the interior of the domain, and three point single sided differencing on the boundaries, except at MAR=6 and MAR=8. The numerical scheme is given in Volume I. Note, that the adiabatic approximation given in Volume I calculates the temperature at boundary points after the energy equation is solved at interior points and at MAR=6 and MAR=8.

```
C*************************************************************************
C      THIS SUBROUTINE CALCULATES THE TEMPERATURE DISTRIBUTION AT EACH X-Y
C      LOCATION AND DEPTH IN THE DOMAIN AT EACH TIME STEP USING A
C      FORWARD DIFFERENCING SCHEME IN TIME
C*************************************************************************
       SUBROUTINE TEMP(IN,JN,KN,T,U,V,W,DX,DY,DZ,DT,H,BH,BV,MAR,TN1,HN1,
      CQRAD,RR,HS,TA,RO)
       DIMENSION T(IN,JN,KN),U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),H(IN,JN)
      C,MAR(IN,JN),TN1(IN,JN,KN),HN1(IN,JN)
      C,RO(IN,JN,KN)
       DO 10 I=1,IN
       DO 10 J=1,JN
       IF(MAR(I,J).EQ.0) GO TO 10
       DO 8 K=2,KN
       IF(MAR(I,J).EQ.11) GO TO 11
       IF(MAR(I,J).EQ.6) GO TO 11
       IF(MAR(I,J).EQ.8) GO TO 11
       IF(MAR(I,J).LT.11) GO To 10
11     CONTINUE
       DHX=(H(I+1,J)-H(I-1,J))/(2.*DX)
       DHY=(H(I,J+1)-H(I,J-1))/(2.*DY)
       D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2.*DX)
       D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2.*DY)
       D2TX=(T(I+1,J,K)+T(I-1,J,K)-2.*T(I,J,K))/(DX*DX)
       D2TY=(T(I,J+1,K)+T(I,J-1,K)-2.*T(I,J,K))/(DY*DY)
       D1HUTX=(H(I+1,J)*U(I+1,J,K)*T(I+1,J,K)-H(I-1,J)*U(I-1,J,K)*
      CT(I-1,J,K))/(2.*DX)
       D1HVTY=(H(I,J+1)*V(I,J+1,K)*T(I,J+1,K)-H(I,J-1)*V(I,J-1,K)*
      CT(I,J-1,K))/(2.*DY)
       IF(K.EQ.5) GO TO 71
       D1WTZ=(W(I,J,K+1)*T(I,J,K+1)-W(I,J,K-1)*T(I,J,K-1))/(2.*DZ)
       D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2.*T(I,J,K))/(DZ*DZ)
       GO TO 80
71     D1WTZ=(3*T(I,J,K)*W(I,J,K)+T(I,J,K-2)*W(I,J,K-2)-4*T(I,J,K-1)*
      CW(I,J,K-1))/(2.*DZ)
       D2TZ=2.*(T(I,J,K-1)-T(I,J,K))/(DZ*DZ)
80     CONTINUE
       TC=(D1HUTX+D1HVTY+H(I,J)*D1WTZ)
       TKX=BH*(DHX*D1TX+H(I,J)*D2TX)
       TKY=BH*(DHY*D1TY+H(I,J)*D2TY)
       TKZ=BV*(D2TZ/H(I,J))
       TN1(I,J,K)=((-TC+TKX+TKY+TKZ)*DT+H(I,J)*T(I,J,K))/HN1(I,J)
8      CONTINUE
10     CONTINUE
       DO 1000 I=1,IN
       DO 1000 J=1,JN
       IF(MAR(I,J).EQ.5) GO To 1000
       IF(MAR(I,J).EQ.6) GO TO 1000
       IF(MAR(I,J).EQ.7) GO TO 1000
       IF(MAR(I,J).EQ.8) GO TO 1000
       IF(MAR(I,J).EQ.9) GO To 1000
       IF(MAR(I,J).EQ.10) GO TO 1000
       IF(MAR(I,J).EQ.11) GO TO 1000
       DO 1001 K=2,KN
       IF(MAR(I,J).EQ.0) TN1(I,J,K)=0.0
       IF(MAR(I,J).EQ.1) TN1(I,J,K)=TN1(I,J-1,K)
```

```
              IF(MAR(I,J).EQ.2) TN1(I,J,K)=TN1(I,J+1,K)
              IF(MAR(I,J).EQ.3) TN1(I,J,K)=TN1(I+1,J,K)
              IF(MAR(I,J).EQ.4) TN1(I,J,K)=TN1(I-1,J,K)
        1001  CONTINUE
        1000  CONTINUE
              DO 2000 I=1,IN
              DO 2000 J=1,JN
              IF(MAR(I,J).EQ.0) GO TO 2000
              IF(MAR(I,J).EQ.1) GO TO 2000
              IF(MAR(I,J).EQ.2) GO TO 2000
              IF(MAR(I,J).EQ.3) GO TO 2000
              IF(MAR(I,J).EQ.4) GO TO 2000
              IF(MAR(I,J).EQ.6) GO TO 2000
              IF(MAR(I,J).EQ.8) GO TO 2000
              IF(MAR(I,J).EQ.11) GO TO 2000
              DO 2001 K=2,KN
              IF(MAR(I,J).EQ.5) TN1(I,J,K)=(TN1(I+1,J,K)+TN1(I,J-1,K))/2.
              IF(MAR(I,J).EQ.7) TN1(I,J,K)=(TN1(I+1,J,K)+TN1(I,J+1,K))/2.
              IF(MAR(I,J).EQ.9) TN1(I,J,K)=(TN1(I-1,J,K)+TN1(I,J+1,K))/2.
              IF(MAR(I,J).EQ.10) TN1(I,J,K)=(TN1(I-1,J,K)+TN1(I,J-1,K))/2.
        2001  CONTINUE
        2000  CONTINUE
              DO 100 I=1,IN
              DO 100 J=1,JN
              IF(MAR(I,J).EQ.0) TN1(I,J,1)=0.0
              RO(I,J,1)=1.029431-.000020*T(I,J,1)-.0000048*(T(    ,1)**2)
              RR=RO(I,J,1)
              TERM=(DZ*HS*HN1(I,J)/(RR*BV))
              TN1(I,J,1)=(TN1(I,J,2)+TA*TERM)/(1.0+TERM)
        100   CONTINUE
              RETURN
              END
```

7.1.18 OLDHT

This subroutine transforms $H(x,y,\sigma)$ at time level n+1 (=HTE(I,J)) into H at time level n (=HTD(I,J)) and H at time level n is transformed to H at time level n-1(=HT(I,J)). These transformations are performed at the end of each time step.

```
C
C**********************************************************************
C     THIS SUBROUTINE TRANSFORMS MATRIX HTE INTO HT FOR THE NEXT TIME CYCL
C     CALCULATION OF THE TOTAL DEPTH AT EACH X-Y LOCATION IN THE DOMAIN
C**********************************************************************
      SUBROUTINE OLDHT(IN,JN,HTE,HTD,HT)
      DIMENSION HTD(IN,JN),HTE(IN,JN),HT(IN,JN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      HT(I,J)=HTD(I,J)
      HTD(I,J)=HTE(I,J)
   10 CONTINUE
      RETURN
      END
```

7.1.19 OLDUV

This subroutine tranforms u and v at time level n=1 (= D (I,J,K) and E(I,J,K)) into u and v at time level n(=H(I,J,K) and G(I,J,K)), and u and v at time level n is transformed into u and v at time level n-1 (=U(I,J,K) and V(I,J,K)). These transformations are performed at the end of each time cycle.

```
C
C*****************************************************************************************
C     THIS SUBROUTINE TRANSFORMS MATRICES D,E INTO U,V, RESPECTIVELY,
C     FOR THE NEXT TIME CYCLE CALCULATION OF HORIZONTAL VELOCITIES AT
C     EACH X-Y LOCATION AND DEPTH IN THE DOMAIN
C*****************************************************************************************
      SUBROUTINE OLDUV(IN,JN,KN,U,V,H,G,D,E)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
     CH(IN,JN,KN),G(IN,JN,KN)
      DO 10 K=1,KN
      DO 10 I=1,IN
      DO 10 J=1,JN
      U(I,J,K)=H(I,J,K)
      V(I,J,K)=G(I,J,K)
      H(I,J,K)=D(I,J,K)
      G(I,J,K)=E(I,J,K)
10    CONTINUE
      RETURN
      END
```

7.1.20 OLDT

This subroutine transforms T at time level n+1 (=TN1(I,J,K)) into T at time level n(=T(I,J,K)). This transformation is performed at the end of each time cycle.

```
C*************************************************************************
C     THIS SUBROUTINE TRANSFORMS MATRIX TN1 INTO T FOR THE NEXT TIME CYCLE
C     CALCULATION OF TEMPERATURE AT EACH X-Y LOCATION AND DEPTH IN THE DOMA
C*************************************************************************
      SUBROUTINE OLDT(IN,JN,KN,T,TN1)
      DIMENSION T(IN,JN,KN),TN1(IN,JN,KN)
      DO 10 K=1,KN
      DO 10 I=1,IN
      DO 10 J=1,JN
      T(I,J,K)=TN1(I,J,K)
10    CONTINUE
      RETURN
      END
```

7.1.21 ETT

This subroutine calculates the wave neight, η (x,y) (=ETA(I,J)) at the end of each time step.

ETA(I,J) = HT(I,J) - HI(I,J)

```
C
C*****************************************************************************
C     THIS SUBROUTINE CALCULATES THE WAVE HEIGHT(SURFACE ELEVATION ABOVE
C     MEAN WATER LEVEL) AT EACH X-Y LOCATION IN THE DOMAIN AT EACH TIME STE
C*****************************************************************************
      SUBROUTINE ETT(IN,JN,HT,HI,MAR,ETA)
      DIMENSION HT(IN,JN),HI(IN,JN),MAR(IN,JN),ETA(IN,JN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      IF(MAR(I,J).EQ.0) GO TO 10
      ETA(I,J)=HT(I,J)-HI(I,J)
10    CONTINUE
      RETURN
      END
```

7.1.22 PRPARA

This subroutine writes out the physical and numerical parameters at the end of each computer run, i.e. after the last time cycle for a particular computer run.

```
C
C*************************************************************************
C      THIS SUBROUTINE PRINTS THE PHYSICAL AND NUMERICAL PARAMETERS FOR THE
C      VARIABLE DENSITY MODEL AT THE END OF EACH COMPUTER RUN
C*************************************************************************
       SUBROUTINE PRPARA(CI,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,
      CRR,KH,KV,BH,BV,QRAD,TI,TTOT1,TA,HS)
       IF(TTOT1.GT.0.) TTOT1=TTOT1-DT
       PRINT 1,CI,CH,CV,CC,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,KH,KV
      C,BH,BV,QRAD,TI,TTOT1,TA,HS
     1 FORMAT(/' CI=',E15.7,/' CH=',E15.7,/' CV=',E15.7/' CP=',E15.7,
      C/' CC=',E15.7,/' DX=',E15.7,/' DY=',E15.7,/' DZ=',E15.7,/' DT=',
      CE15.7,/' TAUX=',E15.7,/' TAUY=',E15.7,/' TTOT=',E15.7,/' GR=',
      CE15.7,/' FF=',E15.7,/' RR=',E15.7,/' KH=', E15.7,/' KV=',E15.7/,
      C/' BH=',E15.7,/' BV=',E15.7,/' QRAD=',E15.7,/' TI=',E15.7,/
      C' TTOT1=',E15.7,/' TA=',E15.7,/' HS =',E15.7/)
       TTOT1=TTOT1+DT
       RETURN
       END
```

7.1.23 PRETA

This subroutine writes out ETA(I,J) at the end of the computer run.

```
C
C***********************************************************************
C     THIS SUBROUTINE PRINTS THE WAVE HEIGHT AT EACH X-Y LOCATION IN THE
C     DOMAIN AT THE END OF EACH COMPUTER RUN
      SUBROUTINE PRETA(I,J,IN,JN,ETA)
      DIMENSION ETA(IN,JN)
      DO 10 I=1,IN
10    PRINT 11,I,(ETA(I,J),J=1,JN)
11    FORMAT(/' I =',I3/' WAVE-HEIGHT'/(5X,8E15.7))
      RETURN
      END
```

7.1.24 PRUV

This subroutine writes out U(I,J,K) and V(I,J,K) at the end of the computer run.

```
C
C*************************************************************************
C*************************************************************************
C      THIS SUBROUTINE PRINTS THE HORIZONTAL VELOCITIES U,V AT EACH X-Y LOC
C      AND DEPTH IN THE DOMAIN AT THE END OF EACH COMPUTER RUN
C*************************************************************************
       SUBROUTINE PRUV(I,J,K,IN,JN,KN,U,V)
       DIMENSION U(IN,JN,KN),V(IN,JN,KN)
       KN1=KN-1
       DO 10 K=1,KN1
       DO 10 I=1,IN
       PRINT 11,K,I,(U(I,J,K),J=1,JN)
 10    PRINT 12,(V(I,J,K),J=1,JN)
 11    FORMAT(/' K=',I3,3X,'I=',I3,/' U-VELOCITY'/(5X,8E15.7))
 12    FORMAT(' V-VELOCITY'/(5X,8E15.7))
       RETURN
       END
```

7.1.25 PRW

This subroutine writes out WZ(I,J,K) at the end of the computer run.

```
C***********************************************************************
C      THIS SUBROUTINE PRINTS THE ACTUAL VERTICAL VELOCITY AT EACH X-Y
C      LOCATION AND DEPTH IN THE DOMAIN AT THE END OF EACH COMPUTER RUN
C***********************************************************************
       SUBROUTINE PRW(IN,JN,KN,W)
       DIMENSION W(IN,JN,KN)
       KN1=KN-1
       DO 10 K=1,KN1
       DO 10 I=1,IN
10     PRINT 11,K,I,(W(I,J,K),J=1,JN)
11     FORMAT(/' K=',I3,3X,' I =',I3,/'  W -VELOCITY'/(5X,8E15.7))
       RETURN
        END
```

7.1.25 PRTEMP

This subroutine writes out T(I,J,K) at the end of the computer run.

```
C**********************************************************************************
C      THIS SUBROUTINE PRINTS THE TEMPERATURE AT EACH X-Y LOCATION AND DEPTH
C      IN THE DOMAIN AT THE END OF EACH COMPUTER RUN
C**********************************************************************************
       SUBROUTINE PRTEMP(I,J,K,IN,JN,KN,T)
       DIMENSION T(IN,JN,KN)
       DO 10 K=1,KN
       DO 10 I=1,IN
       PRINT 11,K,I,(T(I,J,K),J=1,JN)
11     FORMAT(/' K=',I3,3X,' I=',I3,/' TEMPERATURE'/(5X,8E15.7))
10     CONTINUE
       RETURN
       END
```

7.1.27 STORE

This subroutine writes on magnetic tape all calculated system variables and numerical parameters, DX, DY, DZ, DT, TTOT and TTOT1.

```
C
C*******************************************************************.
C      THIS SUBROUTINE WRITES ON MAGNETIC TAPE, FOR THE VARIABLE DENSITY
C      MODEL, THE VALUES FOR THE VARIABLES AND PHYSICAL AND NUMERICAL
C      PARAMETERS FOR STORAGE AND FOR READING IN DATA FOR THE NEXT COMPUTE
C*******************************************************************
      SUBROUTINE STORE(IN,JN,KN,U,V,W,HI,HT,HTD,HX,HY,MAR,ETA,P,RO,CI,
     CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TTOT1,WZ)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),P(IN,JN,KN),
     CHI(IN,JN),HT(IN,JN),HTD(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),
     CETA(IN,JN),RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN)
     C,T(IN,JN,KN),WZ(IN,JN,KN)
      WRITE (8)(((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((H(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((G(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((P(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C((HTD(I,J),J=1,JN),I=1,IN),
     C((HTE(I,J),J=1,JN),I=1,IN),
     C((HI(I,J),J=1,JN),I=1,IN),
     C((HX(I,J),J=1,JN),I=1,IN),
     C((HY(I,J),J=1,JN),I=1,IN),
     C((MAR(I,J),J=1,JN),I=1,IN),
     C((HT(I,J),J=1,JN),I=1,IN),
     C((ETA(I,J),J=1,JN),I=1,IN),
     C(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((WZ(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     CCI,CC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,TTOT1
      END FILE 8
      RETURN
      END
```

7.2 MAIN PROGRAM FOR NASUM III

7.2.1. TMAIN3

This is the main program for free-surface complete field model. This program reads in the data, initializes the necessary quantities, co-ordinates the subroutines and calculates the velocity and temperatures in the whole domain under consideration. The parameter statement defines the size of the computational domain. The subroutine "XYSH" does horizontal stretching. The subroutine "READ 2" reads the MAR matrics which distinguishes the various points in the domain. The subroutine "INITIB" sets the initial conditions on velocities, temperatures, surface height and reads the depths at various points in the domain. The subroutine "CURNT" which is called after "INITIB" sets the velocities everywhere in the domain equal to the current velocity. The subroutine "INLET" puts the discharge velocity and temperature at the discharge location. Then it follows a set of subroutines to calculate the velocity and temperature field for the entire domain. The values of variable at different time levels are given in the Table (1).

```
      PARAMETER IN=20,JN=20,KN=5
      REAL KH,KV,RH,BV
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),UM(IN,JN,KN),
     CHT(IN,JN),HI(IN,JN),ETA(IN,JN),HTE(IN,JN),VM(IN,JN,KN),
     CXX(IN),YY(JN),XXX(IN),YYY(JN),X(IN),Y(JN),A(IN),B(JN),
     CH(IN,JN,KN),G(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),WH(IN,JN,KN),
     CMAR(IN,JN),HTD(IN,JN),PO(IN,JN,KN),P(IN,JN,KN)
     C,T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN),TAM(IN,JN,KN)
      READ 1,IRUN
      READ 1,LN
      READ 1,KSTORE
    1 FORMAT (I5)
      READ 2, CI,CC,CP,CH,CV
      READ 2, GR,FF,RR,HK
      READ 2, DX,DY,DZ
      READ 2, KH,KV,BH,BV
      READ 2,DELX,DELY,DEEX,DEEY,EEEX,EEEY
      CALL XYSH(IN,JN,DELX,DELY,DEEX,DEEY,EEEX,EEEY,XX,YY,XXX,YYY,A,B,
     CX,Y)
    2 FORMAT(   )
      IF(IRUN.GT.0) GO TO 4
      CALL READ2(IN,JN,MAR)
      CALL INITIB(IN,JN,KN,ETA,HT,HI,U,V,RO,P,GR,PR,DZ,HTD,D,E,H,G,
     CHTE,T,TN,TF,TAM,UM,VM,W,DX,DY)
      CALL CURNT(I,J,K,IN,JN,KN,U,V,H,G,D,E)
      CALL INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
      CALL WHTON(IN,JN,KN,HTE,HTD,HTD,ETA,H,G,W,WH,DX,DY,DZ,DT,
     CMAR,XX,YY)
      TTOT=0.0
      GO TO 6
    4 CONTINUF
      CALL READ1(IN,JN,KN,U,V,W,HI,HT,HTD,MAR,ETA,P,RO,CI,UM,VM,
     CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TN,TF,TAM,TAIR,D,
     CE)
    6 CONTINUE
      RFAD 2,TAUX,TAUY
      READ 2,TAIR
      READ 2,DT
      DO 5 L=1,LN
      TTOT=TTOT+DT
      CALL UVEL3(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
     CHTD,HTD,HTE,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,
     CCP,CC,CI,CH,CV,RO,TN,XX,YY,XXX,YYY)
      CALL WVEL(IN,JN,KN,H,G,W,HTD,DX,DY,DZ,MAR,XX,YY,WH)
      CALL TEMS(IN,JN,KN,HTD,HTD,HTE,DX,DY,DZ,DT,BH,BV,T,TN,TF,
     CW,H,G,MAR,HK,TAIR,TAM,RO,XX,YY,XXX,YYY,L,LN)
      CALL BOUND2(I,J,K,IN,JN,KN,U,V,H,G,D,E,W,HI,HTE,T,TN,TF)
      CALL DENSTY(IN,JN,KN,RO,TF)
      CALL PRES(IN,JN,KN,HTE,RO,GR,P,DZ)
      CALL ETT(IN,JN,HTF,HI,MAR,ETA)
      CALL OLDH(IN,JN,HTE,HTD,HT)
      CALL OLDUVT(IN,JN,KN,U,V,H,G,D,F,T,TN,TF)
      CALL INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
      CALL MIXT(I,J,K,IN,JN,KN,TN)
      CALL INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
      CALL TIDE(I,J,K,IN,JN,KN,U,V,H,G,D,E,T,TN,TF)
    5 CONTINUE
      IF(KSTORE.GT.0)GO TO 1000
      CALL STORE(IN,JN,KN,U,V,W,HI,HT,HTD,MAR,ETA,P,RO,CI,UM,VM,CC,CH,
     CCV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TN,TF,TAM,TAIR,D,E)
 1000 CONTINUE
      CALL PRPARA(CI,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,
     CKH,KV,DH,DV,TAIR)
      CALL PRETA(I,J,IN,JN,ETA)
      CALL PRUV(I,J,K,IN,JN,KN,H,G)
      CALL PRW(IN,JN,KN,W)
      CALL PRTEM(IN,JN,KN,TN)
      CALL PRINTP(IN,JN,KN,P)
      CALL PRINTH(IN,JN,HI)
      STOP
      END
```

SUBROUTINE PROGRAMS FOR NASUM III

7.2.2. BOUND2

This subroutine sets the boundary conditions for all variables. The boundary conditions for velocity are no slip and no normal velocity on the bottom and on the shore line (y-axis). For temperature, the bottom and shore line are treated as adiabatic. For oper boundaries, the boundary conditions on velocity and temperature are $\frac{\partial v}{\partial n} = 0$ and $\frac{\partial T}{\partial n} = 0$ respectively. The temperature boundary condition on the surface is specified in temperature subroutine "TEM5".

```
      SUBROUTINE BOUND2(I,J,K,IN,JN,KN,U,V,H,G,D,E,W,HI,HTE,T,TN,TF)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),
     CG(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),T(IN,JN,KN),
     CTN(IN,JN,KN),TF(IN,JN,KN),W(IN,JN,KN),HI(IN,JN),HTE(IN,JN)
      KN1=KN-1
      IN1=IN-1
      JN1=JN-1
      DO 5 K=1,KN
      DO 5 I=1,IN
C
C      FAR J BOUNDARY
C
      T(I,JN,K)=T(I,JN1,K)
      TN(I,JN,K)=TN(I,JN1,K)
      TF(I,JN,K)=TF(I,JN1,K)
      HTE(I,JN)=HTE(I,JN1)
      D(I,JN,K)=D(I,JN1,K)
      E(I,JN,K)=E(I,JN1,K)
      W(I,JN,K)=W(I,JN1,K)
5     CONTINUE
C
C     FAR I BOUNDARY
C
      DO 10 K=1,KN
      DO 10 J=1,JN
      T(IN,J,K)=T(IN1,J,K)
      TN(IN,J,K)=TN(IN1,J,K)
      TF(IN,J,K)=TF(IN1,J,K)
      HTE(IN,J)=HTE(IN1,J)
      D(IN,J,K)=D(IN1,J,K)
      E(IN,J,K)=E(IN1,J,K)
      W(IN,J,K)=W(IN1,J,K)
10    CONTINUE
C
C     ALONG I BOUNDARY
C
      DO 15 K=1,KN
      DO 15 I=1,IN
      T(I,1,K)=T(I,2,K)
      TN(I,1,K)=TN(I,2,K)
      TF(I,1,K)=TF(I,2,K)
      HTE(I,1)=HTE(I,2)
      D(I,1,K)=D(I,2,K)
      E(I,1,K)=E(I,2,K)
      W(I,1,K)=W(I,2,K)
15    CONTINUE
C
C      ALONG SHORE LINE
C
      DO 20 K=1,KN
      DO 20 J=1,JN
      T(1,J,K)=T(2,J,K)
      TN(1,J,K)=TN(2,J,K)
      TF(1,J,K)=TF(2,J,K)
      HTE(1,J)=HI(1,J)
      D(1,J,K)=D(2,J,K)
```

```
         E(1,J,K)=E(2,J,K)
         W(1,J,K)=W(2,J,K)
   20    CONTINUE
   C
   C     BOTTOM BOUNDARY CONDITION
   C
         DO 25 I=1,IN
         DO 25 J=1,JN
         T(I,J,5)=T(I,J,4)
         TN(I,J,5)=TN(I,J,4)
         TF(I,J,5)=TF(I,J,4)
   25    CONTINUE
         RETURN
         END
```

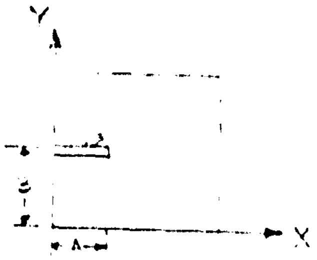
7.2.3. CONST

This is a small main program and need to be run in order to determine the constants DEEX, DEEY, EEEX, EEEY which are used in the subroutine "XYSH". The input cards reads

(1) XB, A, DX, AN

(2) YB, B, DY, BN

where XB, YB are X and Y boundary distances. A and B are distances from the shore line (y-axis) and x-axis respectively from where the stretching starts. It is shown below.



DX, DY are the minimum grid size needed. AN, BN are the number of grid points in the x and y directions.

```
C
   50 READ 4,XB,A,DX,AN
    4 FORMAT(   )
      IF(XB.LE.0.) GO TO 400
      WRITE(6,11) XB,A,DX,AN
   11 FORMAT('1','XBNDRY=',F10.1,'    A=',F10.1,'    DELTA X=',F10.1,'
     1NBR POINTS=',F10.1/)
      E1=XB-A
      E2=(AN-1.)*DX
      DC=.01*A
   18 C=0.
      J=0
    8 C=C+DC
      J=J+1
      D=C*ALOG(A/C+SQRT((A/C)**2+1.))
      U=AMIN1((E2-D)/C,85.)
      X=A+C*SINH(U)
      WRITE(6,42) C,X
      IF(X.GT.XB) GO TO 8
      IF(J.GT.1) GO TO 17
      DC=DC/2.
      GO TO 18
   17 CMIN=C-DC
      CMAX=C
   33 CONTINUE
C
      C11=CMAX
      D=C11*ALOG(A/C11+SQRT((A/C11)**2+1.))
      U=AMIN1((E2-D)/C11,85.)
      ERR1=-E1+C11*SINH(U)
      J=1
      C12=CMAX-DC/2.
C
    1 J=J+1
      IF(J.GT.30) GO TO 99
      D=C12*ALOG(A/C12+SQRT((A/C12)**2+1.))
      U=AMIN1((E2-D)/C12,85.)
      ERR2=-E1+C12*SINH(U)
      WRITE(6,42) C12,D,U,ERR2
   42 FORMAT(1X,4F15.7)
      IF(ABS(ERR2/XB).LT..001) GO TO 2
      C13=(C11*ERR2-C12*ERR1)/(ERR2-ERR1)
      C13=AMAX1(C13,CMIN)
      C13=AMIN1(C13,CMAX)
      C11=C12
      C12=C13
      ERR1=ERR2
      GO TO 1
C
    2 C1=C12
      WRITE(6,6) C1,D
    6 FORMAT(1H0,'C1.',F15.7,'    D=',E15.7/)
      DUM=0.
      N=INT(AN)
      DO 3 I=1,N
      XL=(I-1)*DX
      X=A+C1*SINH((XL-D)/C1)
      DELTA=X-DUM
      DUM=X
      WRITE(6,5) I,XL,X,DELTA
    3 CONTINUE
    5 FORMAT(1X,'I=',I4,'   XL=',F10.2,'   X=',F10.2,'   DELTA=',F10.1)
      GO TO 50
   99 WRITE(6,14)
   14 FORMAT(1X,'NBR ITERATIONS EXCEEDED 30')
      GO TO 400
  400 STOP
      END
```

7.2.4. CURNT

This subroutine sets the velocity field in the whole domain equal to the current velocity. In this case 2 cm/sec is chosen. If the initial current is more, the values should be made equal to the measured value of current. If there is no initial current, this subroutine can be deleted from the main program.

```
      SUBROUTINE CURNT(I,J,K,IN,JN,KN,U,V,H,G,D,E)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN)
     C,H(IN,JN,KN),G(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
      KN1=KN-1
      DO 10 K=1,KN1
      DO 10 I=1,IN
      DO 10 J=1,JN
      U(I,J,K)=0.0
      V(I,J,K)=2.0
      H(I,J,K)=0.0
      G(I,J,K)=2.0
      D(I,J,K)=0.0
      E(I,J,K)=2.0
   10 CONTINUE
      RETURN
      END
```

7.2.5. DENSTY

This subroutine computes for the whole domain the density using the temperatures computed in the subroutine "TEM5".

```
      SUBROUTINE DENSTY(IN,JN,KN,RO,T)
      DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      DO 10 K=1,KN
      RO(I,J,K)=1.000428-0.000019*(T(I,J,K))-0.0000046*
     C(T(I,J,K)*T(I,J,K))
   10 CONTINUE
      RETURN
      END
```

7.2.6. ETT

This subroutine calculates the wave height by subtracting mean level of water (HI) from the total height computed.

```
C      THIS PROGRAM COMPUTES THE WAVE-HEIGHT
C
       SUBROUTINE ETT(IN,JN,HT,HI,MAR,ETA)
       DIMENSION HT(IN,JN),HI(IN,JN),MAR(IN,JN),ETA(IN,JN)
       DO 10 I=1,IN
       DO 10 J=1,JN
       ETA(I,J)=HT(I,J)-HI(I,J)
10     CONTINUE
       RETURN
       END
```

7.2.7. HITE2

This subroutine computes the free surface height using the equation

$$\frac{\partial H}{\partial t} = - \int_{o}^{1} \left(X'\frac{\partial(HU)}{\partial X} + Y'\frac{\partial(HV)}{\partial Y}\right)\delta\sigma \quad -\left(W_b - U_b\, X'\frac{\partial h}{\partial X}\right.$$

$$\left. - V_b Y'\frac{\partial h}{\partial Y}\right)$$

The numerical scheme used is forward difference in time and central difference in space (FTCS). Simpsons rule is used for numerical integration.

```
      SUBROUTINE HITE2(IN,JN,KN,MAR,U,V,HT,HTD,HTE,DZ,DT,DX,DY,HDUM
     C,XX,YY,WH)
      DIMENSION MAR(IN,JN),U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),HTD(IN,JN),
     CHDUM(IN,JN),HTE(IN,JN),XX(IN),YY(JN),WH(IN,JN,KN)
      KNM1=KN-1
      IN1=IN-1
      JN1=JN-1
      DO 50 I=2,IN1
      DO 50 J=2,JN1
      HDUM(I,J)=0.0
      DO 60 K=1,KN
      IF(MAR(I,J).EQ.11) GO TO 11
      GO TO 50
   11 DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
      DIHVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
C....SIMPSON'S RULE IS USED FOR INTEGRATION
      IF(K.EQ.1.OR.K.EQ.5) GO TO 101
      IF(K.EQ.2.OR.K.EQ.4) GO TO 102
      HDUM(I,J)=((DIHUX*XX(I)+DIHVY*YY(J))*DZ*(2./3.))+HDUM(I,J)
      GO TO 103
  101 HDUM(I,J)=((DIHUX*XX(I)+DIHVY*YY(J))*DZ/3.)+HDUM(I,J)
      GO TO 103
  102 HDUM(I,J)=((DIHUX*XX(I)+DIHVY*YY(J))*DZ*(4./3.))+HDUM(I,J)
  103 CONTINUE
      HTE(I,J)=HTD(I,J)-HDUM(I,J)*DT
   60 CONTINUE
      HTE(I,J)=HTE(I,J)-WH(I,J,KN)*DT
   50 CONTINUE
      RETURN
      END
```

7.2.8. INITIB

This subroutine reads the depths for a constant depth basin and initializes the values u, v, w, p, ρ, T and HI. The program sets u, v, w and wave height ETA equal to zero. The temperature is set equal to the ambient temperature. The pressure is hydrostatic.

```
      SUBROUTINE INITIB(IN,JN,KN,ETA,HT,HI,U,V,RO,P,GR,RR,DZ,HTD,D,
     CE,H,G,HTE,T,TN,TF,TAM,UM,VM,W,DX,DY)
      DIMENSION ETA(IN,JN),HT(IN,JN),HI(IN,JN),U(IN,JN,KN),V(IN,JN,KN),
     CRO(IN,JN,KN),P(IN,JN,KN),HTD(IN,JN),W(IN,JN,KN)
     C,D(IN,JN,KN),E(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN)
     C,TAM(IN,JN,KN),T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
     C,UM(IN,JN,KN),VM(IN,JN,KN)
      DO 500 J=1,JN
      HI(1,J)=300.0
      HI(2,J)=350.0
      HI(3,J)=400.0
      HI(4,J)=450.0
      HI(5,J)=500.0
      HI(6,J)=550.0
      HI(7,J)=600.0
      HI(8,J)=600.0
      HI(9,J)=600.0
      HI(10,J)=600.0
      HI(11,J)=600.0
      HI(12,J)=650.0
      HI(13,J)=700.0
      HI(14,J)=750.0
      HI(15,J)=800.0
      HI(16,J)=850.0
      HI(17,J)=900.0
      HI(18,J)=950.0
      HI(19,J)=1000.0
      HI(20,J)=1000.0
  500 CONTINUE
      DO 10 I=1,IN
      DO 10 J=1,JN
      ETA(I,J)=0.0
      HT(I,J)=HI(I,J)
      HTD(I,J)=HI(I,J)
      HTE(I,J)=HI(I,J)
   10 CONTINUE
      DO 8 I=1,IN
      DO 8 J=1,JN
      DO 8 K=1,KN
      UM(I,J,K)=0.0
      VM(I,J,K)=0.0
      W(I,J,K)=0.0
      U(I,J,K)=UM(I,J,K)
      V(I,J,K)=VM(I,J,K)
      TAM(I,J,K)=25.0
      T(I,J,K)=TAM(I,J,K)
      P(I,J,K)=GR*HT(I,J)*RR*(K-1)*DZ
      H(I,J,K)=U(I,J,K)
      D(I,J,K)=U(I,J,K)
      G(I,J,K)=V(I,J,K)
      E(I,J,K)=V(I,J,K)
      TN(I,J,K)=T(I,J,K)
      TF(I,J,K)=T(I,J,K)
    8 CONTINUE
      RETURN
      END
```

7.2.9. INLET

This subroutine puts the velocity and temperature of the discharge at the discharge location. The value of velocity specif'ed in this subroutine should be calculated depunding upon the mass of the discharge.

```
      SUBROUTINE INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
      DIMENSION T(IN,JN,KN),WH(IN,JN,KN)
     C,TN(IN,JN,KN),TF(IN,JN,KN)
      DO 10 I=8,10
      DO 10 J=10,12
      DO 10 K=1,KN
      WH(I,J,5)=-0.35
      T(I,J,K)=35.0
      TN(I,J,K)=35.0
      TF(I,J,K)=35.0
   10 CONTINUE
      RETURN
      END
```

7.2.10. MIXT

This subroutine mixes the temperatures by an averaging process in such that unstable density gradients are eliminated.

```
      SUBROUTINE MIXT(I,J,K,IN,JN,KN,T)
      DIMENSION T(IN,JN,KN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      K=1
      IF (T(I,J,K).GE.T(I,J,K+1)) GO TO 1
       AVT=(T(I,J,K)+T(I,J,K+1))/2.0
      T(I,J,K)=AVT
      T(I,J,K+1)=AVT
    1 CONTINUE
      IF (T(I,J,K+1).GE.T(I,J,K+2)) GO TO 2
      AVT=(T(I,J,K)+T(I,J,K+1)+T(I,J,K+2))/3.0
      T(I,J,K)=AVT
      T(I,J,K+1)=AVT
      T(I,J,K+2)=AVT
    2 CONTINUE
      IF (T(I,J,K+2).GE.T(I,J,K+3)) GO TO 3
      AVT=(T(I,J,K)+T(I,J,K+1)+T(I,J,K+2)+T(I,J,K+3))/4.0
      T(I,J,K)=AVT
      T(I,J,K+1)=AVT
      T(I,J,K+2)=AVT
      T(I,J,K+3)=AVT
    3 CONTINUE
      IF (T(I,J,K+3).GE.T(I,J,K+4)) GO TO 4
      AVT=(T(I,J,K)+T(I,J,K+1)+T(I,J,K+2)+T(I,J,K+3)+T(I,J,K+4))/5.0
      T(I,J,K)=AVT
      T(I,J,K+1)=AVT
      T(I,J,K+2)=AVT
      T(I,J,K+3)=AVT
      T(I,J,K+4)=AVT
    4  CONTINUE
   10 CONTINUE
      RETURN
      END
```

7.2.11. OLDHT

This subroutine sets the values of height HTD at time level n to HT at time level n-1 and HTE at time level n+1 to HTD at time level n after all computation are completed. This is necessary in order to retain values of height at one time step lag.

```
C      THIS PROGRAM  TRANSFERS MATRIX HTD TO HT
C
       SUBROUTINE OLDHT(IN,JN,HTE,HTD,HT)
       DIMENSION HTD(IN,JN),HTE(IN,JN),HT(IN,JN)
       DO 10 I=1,IN
       DO 10 J=1,JN
       HT(I,J)=HTD(I,J)
       HTD(I,J)=HTE(I,J)
   10  CONTINUE
       RETURN
       END
```

7.2.12. OLDUVT

This subroutine sets the values of velocity and temperature at time level n+l equal to the values at time level n and the values at time level n are made equal to the values at time level n-l. This is necessary in order to retain the values of velocity and temperature at the time step lag.

```
C     THIS PROGRAM TRANSFERS MATRICES U,V TO D,E RESPECTIVELY
C     AND T TO TN
C
      SUBROUTINE OLDUVT(IN,JN,KN,U,V,H,G,D,E,T,TN,TF)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
     CH(IN,JN,KN),G(IN,JN,KN),T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
      DO 10 K=1,KN
      DO 10 I=1,IN
      DO 10 J=1,JN
      U(I,J,K)=H(I,J,K)
      V(I,J,K)=G(I,J,K)
      H(I,J,K)=D(I,J,K)
      G(I,J,K)=E(I,J,K)
      T(I,J,K)=TN(I,J,K)
      TN(I,J,K)=TF(I,J,K)
   10 CONTINUE
      RETURN
      END
```

7.2.13. PRPARA

This subroutine prints the input values and the total time the model is simulated.

```
C      THIS SUBROUTINE PRINTS PARAMETERS FOR THE FREE SURFACE MODEL
C
       SUBROUTINE PRPARA(CI,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,
      CRR,KH,KV,BH,BV,TAIR)
       PRINT 1,CI,CH,CV,CC,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,KH,KV
      C,BH,BV,TAIR
 1     FORMAT(/' CI=',F15.7,/' CH=',E15.7,/' CV=',E15.7/' CP=',E15.7,
      C/' CC=',E15.7,/' DX=',E15.7,/' DY=',E15.7,/' DZ=',E15.7,/' DT=',
      CE15.7,/' TAUX=',E15.7,/' TAUY=',E15.7,/' TTOT=',E15.7,/' GR=',
      CE15.7,/' FF=',E15.7,/' RR=',E15.7,/' KH=', E15.7,/' KV=',E15.7,/
      C' BH=',E15.7,/' BV=',F15.7,/' TAIR=',F15.7/)
       RETURN
       END
```

7.2.14. PRES

This subroutine calculates the pressure field using the updated density field.

```
C     THIS PROGRAM CALCULATES THE PRESSURE FIELD
C
      SUBROUTINE PRES(IN,JN,KN,HT,RO,GR,P,DZ)
      DIMENSION HT(IN,JN),RO(IN,JN,KN),P(IN,JN,KN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      P(I,J,1)=0.0
      DO 8 K=2,KN
      P(I,J,K)=P(I,J,K-1)+GR*HT(I,J)*(RO(I,J,K-1)+RO(I,J,K))*DZ/2.0
8     CONTINUE
10    CONTINUE
      RETURN
      END
```

7.2.15. PRETA

This subroutine prints wave height (ETA).

```
C      THIS PROGRAM PRINTS THE WAVE HEIGHT
C
       SUBROUTINE PRETA(I,J,IN,JN,ETA)
       DIMENSION ETA(IN,JN)
       DO 10 I=1,IN
10     PRINT 11,I,(ETA(I,J),J=1,JN)
11     FORMAT(/' I =',I3/' WAVE-HEIGHT'/(5X,8E15.7))
       RETURN
       END
```

7.2.16. PRTEM

This subroutine prints temperatures in the whole domain.

```
C     THIS SUBROUTINE PRINTS THE TEMPERATURES
C
      SUBROUTINE PRTEM(IN,JN,KN,T)
      DIMENSION T(IN,JN,KN)
      DO 40 K=1,KN
      WRITE(6,105) K
 105  FORMAT('1','   TEMPERATURE AT K =',I5,//)
      DO 20 I=1,IN
  20  WRITE(6,106) (T(I,J,K),J=1,JN)
 106  FORMAT(/,20F6.2)
  40  CONTINUE
      RETURN
      END
```

7.2.17. PRUV

This subroutine prints u and v velocity in the whole domain.

```
C      THIS PROGRAM PRINTS THE HORIZONTAL VELOCITIES
C
       SUBROUTINE PRUV(I,J,K,IN,JN,KN,U,V)
       DIMENSION U(IN,JN,KN),V(IN,JN,KN)
       KN1=KN-1
       DO 10 K=1,KN
       DO 10 I=1,IN
       PRINT 11,K,I,(U(I,J,K),J=1,JN)
10     PRINT 12,(V(I,J,K),J=1,JN)
11     FORMAT(/' K=',I3,3X,'I=',I3,/' U-VELOCITY'/(5X,8E15.7))
12     FORMAT(' V-VELOCITY'/(5X,8E15.7))
       RETURN
       END
```

7.2.18. PRW

This subroutine prints Ω values in the whole domain.

```
C      THIS SUBROUTINE PRINTS THE VERTICAL VFLOCITIES
C
       SUBROUTINE PRW(IN,JN,KN,W)
       DIMENSION W(IN,JN,KN)
       DO 10 K=1,KN
       DO 10 I=1,IN
10     PRINT 11,K,I,(W(I,J,K),J=1,JN)
11     FORMAT(/' K=',I3,3X,' I =',I3,/'  W -VELOCITY'/(5X,8E15.7))
       RETURN
        END
```

7.2.19. PRWH

This subroutine prints w-velocity in the whole domain.

```
      SUBROUTINE PRWH(IN,JN,KN,WH)
      DIMENSION WH(IN,JN,KN)
      DO 10 K=1,KN
      DO 10 I=1,IN
10    PRINT 11,K,I,(WH(I,J,K),J=1,JN)
11    FORMAT(/' K=',I3,3X,' I =',I3,/'  WH-VELOCITY'/(5X,8E15.7))
      RETURN
       END
```

7.2.20. READ1

This subroutine is used to read the information stored by subroutine "STORE". This subroutine is used from second run onwards in order to use the values created in the first run (ie IRUN=0). "READ1" and "STORE" correspond to each other. The "READ 1" subroutine uses a file designated as "UNIT 7" in order to read the information on the tape.

```
C     THIS SUBROUTINE READS DATA FROM TAPE
C
      SUBROUTINE READ1(IN,JN,KN,U,V,W,HI,HT,HTD,MAR,ETA,P,RO,CI,UM,VM,
     CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TN,TF,TAM,TAIR,D,
     CE)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),P(IN,JN,KN),
     CHI(IN,JN),HT(IN,JN),HTD(IN,JN),MAR(IN,JN),
     CETA(IN,JN),RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN),
     CT(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN),TAM(IN,JN,KN)
     C,UM(IN,JN,KN),VM(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
1     CONTINUE
      READ(7,END=1)  (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((H(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((G(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((P(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((UM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((VM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C((HTD(I,J),J=1,JN),I=1,IN),((HTE(I,J),J=1,JN),I=1,IN),
     C((HI(I,J),J=1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),
     C((HT(I,J),J=1,JN),I=1,IN),((ETA(I,J),J=1,JN),I=1,IN),
     C(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((TN(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((TF(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((TAM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     CCI,CC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,TAIR
      RETURN
      END
```

7.2.21. READ2

This subroutine reads the "MAR" matrix. The MAR numbering system is used in order to identify the points in the interior, on the boundaries and outside the domain. The "MAR" numbering used is as follows,

MAR (I,J) = 0 for points outside the domain.

MAR (I,J) = 1 for upper horizontal boundary.

MAR (I,J) = 2 for lower horizontal boundary.

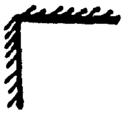
MAR (I,J) = 3 for left vertical boundary.

MAR (I,J) = 4 for right vertical boundary.

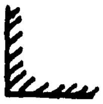
MAR (I,J) = 5 through MAR (I,J) = 10 are boundary corners and are specified as below.

MAR (I,J) = 11 for points in the iterior.

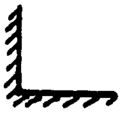
MAR=5



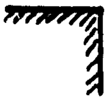
MAR=6



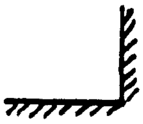
MAR=7



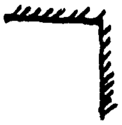
MAR=8



MAR=9



MAR=10



```
C     THIS SUBROUTINE DEFINES THE MAR MARTRIX FOR LOCATING THE POSITIONS
C
      SUBROUTINE READ2(IN,JN,MAR)
      DIMENSION MAR(IN,JN)
      MAR(1,1)=7
      MAR(1,JN)=5
      MAR(IN,1)=9
      MAR(IN,JN)=10
      INM1=IN-1
      JNM1=JN-1
      DO 10 I=2,INM1
      MAR(I,1)=2
   10 MAR(I,JN)=1
      DO 20 J=2,JNM1
      MAR(1,J)=3
   20 MAR(IN,J)=4
      DO 30 I=2,INM1
      DO 30 J=2,JNM1
   30 MAR(I,J)=11
      RETURN
      END
```

7.2.22. STORE

This subroutine is used to store the values at the end of all computations on a file designated as "UNIT 8".

```
C          THIS SUBROUTINE STORES THE DATA INTO TAPE
C
            SUBROUTINE STORE(IN,JN,KN,U,V,W,HI,HT,HTD,MAR,ETA,P,RO,CI,UM,VM,
           CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TN,TF,TAM,TAIR,
           CD,E)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),P(IN,JN,KN),
           CHT(IN,JN),HI(IN,JN),HTD(IN,JN),MAR(IN,JN),D(IN,JN,KN),E(IN,JN,KN),
           CETA(IN,JN),RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN),
           CT(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN),TAM(IN,JN,KN)
           C,UM(IN,JN,KN),VM(IN,JN,KN)
            WRITE (8)(((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((H(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((G(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((P(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((UM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((VM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C((HTD(I,J),J=1,JN),I=1,IN),((HTE(I,J),J=1,JN),I=1,IN),
           C((HI(I,J),J=1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),
           C((HT(I,J),J=1,JN),I=1,IN),((ETA(I,J),J=1,JN),I=1,IN),
           C(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((TN(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((TF(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((TAM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           CCI,CC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,TAIR
            END FILE 8
            END FILE 8
            RETURN
            END
```

7.2.23. TEM5

This subroutine computes temperature only in the interior of the domain. The schemes used are forward in time and central in space (F.T.C.S.). Vertical diffusion term is treated by DuFort-Frankel scheme.

```
      SUBROUTINE TEMS(IN,JN,KN,HT,HTD,HTE,DX,DY,DZ,DT,RH,BV,T,TN,TF,
     CW,H,G,MAR,HK,TAIR,TAM,RO,XX,YY,XXX,YYY,L,LN)
      DIMENSION HT(IN,JN),HTD(IN,JN),HTE(IN,JN),MAR(IN,JN),
     CTAM(IN,JN,KN),W(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),RO(IN,JN,KN),
     CXX(IN),YY(JN),XXX(IN),YYY(JN),
     CT(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
      KN1=KN-1
      IN1=IN-1
      JN1=JN-1
      DO 10 K=1,KN1
      DO 10 I=2,IN1
      DO 10 J=2,JN1
      IF(MAR(I,J).EQ.11) GO TO 11
      GO TO 300
11    CONTINUE
      DHUTX=(HTD(I+1,J)*H(I+1,J,K)*TN(I+1,J,K)-HTD(I-1,J)
     C*H(I-1,J,K)*TN(I-1,J,K))/(2*DX)
      DHVTY=(HTD(I,J+1)*G(I,J+1,K)*TN(I,J+1,K)-
     CHTD(I,J-1)*G(I,J-1,K)*TN(I,J-1,K))/(2*DY)
      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
      D1TY=(TN(I,J+1,K)-TN(I,J-1,K))/(2*DY)
      D2TY=(TN(I,J+1,K)+TN(I,J-1,K)-2*TN(I,J,K))/(DY*DY)
      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
      D1TX=(TN(I+1,J,K)-TN(I-1,J,K))/(2*DX)
      D2TX=(TN(I+1,J,K)+TN(I-1,J,K)-2*TN(I,J,K))/(DX*DX)
      IF(K.EQ.1) GO TO 50
      D1WTZ=(W(I,J,K+1)*TN(I,J,K+1)-W(I,J,K-1)*TN(I,J,K-1))/(2*DZ)
      D1TZ=(TN(I,J,K+1)-TN(I,J,K-1))/(2*DZ)
      D2TZ1=(TN(I,J,K+1)+TN(I,J,K-1)-T(I,J,K))/(DZ*DZ)
      GO TO 200
50    D1WTZ=(4*W(I,J,K+1)*TN(I,J,K+1)-3*W(I,J,K)*TN(I,J,K)-
     CW(I,J,K+2)*TN(I,J,K+2))/(2*DZ)
      D1TZ=HT(I,J)*HK*(TN(I,J,1)-TAIR)
      D2TZ1=(2*TN(I,J,K+1)-T(I,J,K))/(DZ*DZ)-2*D1TZ/DZ
200   CONTINUE
      DHT=(HTE(I,J)-HTD(I,J))/(DT)
      TLX=DHUTX*XX(I)
      TLY=DHVTY*YY(J)
      TLZ=HTD(I,J)*D1WTZ
      TLC=((K-1)*DZ-1)*DHT*D1TZ
      TL=TLX+TLY+TLZ+TLC
      TRX=RH*(DHX*XX(I)*D1TX*XX(I)+HT(I,J)*XX(I)*XX(I)*D2TX+HT(I,J)*
     CXXX(I)*D1TX)
      TRY=RH*(DHY*YY(J)*D1TY*YY(J)+HT(I,J)*YY(J)*YY(J)*D2TY+HT(I,J)*
     CYYY(J)*D1TY)
      GO TO 500
500   TR=TRX+TRY
      TRV=BV*D2TZ1/HTD(I,J)
      TR=TR+TRV
      TF(I,J,K)=((TR-TL)*DT+HTD(I,J)*TN(I,J,K))/
     C(HTE(I,J)+BV*DT/(DZ*DZ*HTD(I,J)))
      GO TO 8
300   TF(I,J,K)=TAM(I,J,K)
8     CONTINUE
10    CONTINUE
      RETURN
      END
```

7.2.24. TIDE

This sets the values of velocity at one boundary equal to the value of current comming into the domain.

```
      SUBROUTINE TIDE(I,J,K,IN,JN,KN,U,V,H,G,D,E,T,TN,TF)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN)
      DIMENSION G(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
      DIMENSION T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
      KN1=KN-1
      DO 10 I=1,IN
      DO 10 K=1,KN1
      V(I,1,K)=2.0
      G(I,1,K)=2.0
      E(I,1,K)=2.0
   10 CONTINUE
      RETURN
      END
```

7.2.25. UVEL3

This subroutine computes u and v velocities from the two horizontal momentum equations. The schemes used are forward in time and central in space. DuFort-Frankel scheme is used on the vertical viscous terms. This subroutine computes velocities only in the interior.

```
      SUBROUTINE UVEL3(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
     CHT,HTD,HTF,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,CP,CC,CI,CH,CV,RO,T,
     CXX,YY,XXX,YYY)
      REAL KH,KV
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),
     CD(IN,JN,KN),E(IN,JN,KN),HT(IN,JN),HTD(IN,JN),HTE(IN,JN),HX(IN,JN),
     CHY(IN,JN),ETA(IN,JN),P(IN,JN,KN),MAR(IN,JN),W(IN,JN,KN)
     C,RO(IN,JN,KN),T(IN,JN,KN),
     CXX(IN),YY(JN),XXX(IN),YYY(JN)
      KN1=KN-1
      IN1=IN-1
      JN1=JN-1
      DO 10 I=2,IN1
      DO 10 J=2,JN1
      DO 9 K=1,KN1
      IF(MAR(I,J).EQ.11) GO TO 11
      GO TO 10
 11   CONTINUE
      ETAY=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
      D1PY=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
      D1HUUY=(H(I+1,J,K)*H(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*
     CH(I-1,J,K)*HTD(I-1,J))/(2*DX)
      D1HUVY=(H(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-H(I,J-1,K)*
     CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
      D1UX=(H(I+1,J,K)-H(I-1,J,K))/(2*DX)
      D2UX=(H(I+1,J,K)+H(I-1,J,K)-2*H(I,J,K))/(DX*DX)
      D1UY=(H(I,J+1,K)-H(I,J-1,K))/(2*DY)
      D2UY=(H(I,J+1,K)+H(I,J-1,K)-2*H(I,J,K))/(DY*DY)
 100  CONTINUE
      IF(K.EQ.1) GO TO 70
      D1UWZ=(H(I,J,K+1)*W(I,J,K+1)-H(I,J,K-1)*W(I,J,K-1))/(2*DZ)
      D1UZ=(H(I,J,K+1)-H(I,J,K-1))/(2*DZ)
      D2UZ1=(H(I,J,K+1)+H(I,J,K-1)-U(I,J,K))/(DZ*DZ)
      GO TO 80
 70   D1UWZ=(4*H(I,J,K+1)*W(I,J,K+1)-3*H(I,J,K)*W(I,J,K)-H(I,J,K+2)*
     CW(I,J,K+2))/(2.*DZ)
      D1UZ=(TAUX*HTD(I,J))/(KV)
      D2UZ1=(2*H(I,J,K+1)-U(I,J,K))/(DZ*DZ)-2*D1UZ/DZ
 80   CONTINUE
      DHT=(HTF(I,J)-HTD(I,J))/(DT)
      UI=CI*(D1HUUX*XX(I)+D1HUVY*YY(J)+HTD(I,J)*D1UWZ+
     C((K-1)*DZ-1)*D1UZ*DHT)
      UP1=CP*HTD(I,J)*(ETAX*GR)*(-1.)*XX(I)
      UP2=-(D1FX*XX(I)/RR)*HTD(I,J)
      UP3=(GR*XX(I)*DHX*(K-1)*DZ)*HTD(I,J)
      UP=UP1+UP2+UP3
      UC=CC*HTD(I,J)*FF*G(I,J,K)
      UH=CH*KH*(DHX*XX(I)*XX(I)*D1UX+HT(I,J)*XX(I)*XX(I)*D2UX
     C+HT(I,J)*XXX(I)*D1UY)+
     CCH*KH*(DHY*YY(J)*YY(J)*D1UY+HT(I,J)*YY(J)*YY(J)*D2UY
     C+HT(I,J)*YYY(J)*D1UY)
      UV=(KV*D2UZ1)/HTD(I,J)
      D(I,J,K)=((-UI+UC+UH+UV)*DT+HTD(I,J)*H(I,J,K))/
     C(HTF(I,J)+(KV*DT)/(DZ*DZ*HTD(I,J)))
```

```
    8 CONTINUE
   10 CONTINUE
      DO 30 I=2,IN1
      DO 30 J=2,JN1
      DO 7 K=1,KN1
      IF(MAP(I,J).EQ.11) GO TO 12
      GO TO 30
   12 CONTINUE
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
      D1PY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
      D1HUVX=(H(I+1,J,K)*G(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*G(I-1,J,K)*
     CHTD(I-1,J))/(2*DX)
      D1HVVY=(G(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-G(I,J-1,K)*
     CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
      D1VX=(G(I+1,J,K)-G(I-1,J,K))/(2*DX)
      D2VX=(G(I+1,J,K)+G(I-1,J,K)-2*G(I,J,K))/(DX*DX)
      D1VY=(G(I,J+1,K)-G(I,J-1,K))/(2*DY)
      D2VY=(G(I,J+1,K)+G(I,J-1,K)-2*G(I,J,K))/(DY*DY)
      IF(K.EQ.1) GO TO 90
      D1VWZ=(G(I,J,K+1)*W(I,J,K+1)-G(I,J,K-1)*W(I,J,K-1))/(2*DZ)
      D1VZ=(G(I,J,K+1)-G(I,J,K-1))/(2*DZ)
      D2VZ1=(G(I,J,K+1)+G(I,J,K-1)-V(I,J,K))/(DZ*DZ)
      GO TO 95
   90 D1VWZ=(4*G(I,J,K+1)*W(I,J,K+1)-3*G(I,J,K)*W(I,J,K)-G(I,J,K+2)*
     CW(I,J,K+2))/(2*DZ)
      D1VZ=(TAUY*HTD(I,J))/(KV)
      D2VZ1=(2*G(I,J,K+1)-V(I,J,K))/(DZ*DZ)-2*D1VZ/DZ
   95 CONTINUE
      DHT=(HTF(I,J)-HTD(I,J))/(DT)
      VI=CI*(D1HUVX*XX(I)+D1HVVY*YY(J)+HTD(I,J)*D1VWZ+
     C((K-1)*DZ-1)*D1VZ*DHT)
      VP1=CP*HTD(I,J)*(ETAY*GR)*(-1.)*YY(J)
      VP2=-(D1PY*YY(J)/RR)*HTD(I,J)
      VP3=(GR*YY(J)*DHY*(K-1)*DZ)*HTD(I,J)
      VP=VP1+VP2+VP3
      VC=CC*HTD(I,J)*FF*H(I,J,K)
      VH=CH*KH*(DHX*XX(I)*XX(I)*D1VX+HT(I,J)*XX(I)*XX(I)*D2VX
     C+HT(I,J)*XXX(I)*D1VX)
     C+CH*KH*(DHY*YY(J)*YY(J)*D1VY+HT(I,J)*YY(J)*YY(J)*D2VY
     C+HT(I,J)*YYY(J)*D1VY)
      VV=KV*D2VZ1/HTD(I,J)
      E(I,J,K)=((-VI+VC+VH+VV)*DT+HTD(I,J)*G(I,J,K))/
     C(HTF(I,J)+(KV*DT)/(DZ*DZ*HTD(I,J)))
    7 CONTINUE
   30 CONTINUE
      RETURN
      END
```

7.2.26. WHTOW

This subroutine calculates the value of $W(\Omega)$ which is used in the model from the value of WH (w) specified.

```
      SUBROUTINE WHTOW(IN,JN,KN,HTD,HTE,HT,ETA,D,E,W,WH,DX,DY,DZ,DT,MAR,
     CXX,YY)
      DIMENSION HT(IN,JN),HTD(IN,JN),HTE(IN,JN),MAR(IN,JN),D(IN,JN,KN),
     CE(IN,JN,KN),W(IN,JN,KN),WH(IN,JN,KN),ETA(IN,JN),XX(IN),YY(JN)
      KN1=KN-1
      IN1=IN-1
      JN1=JN-1
      DO 10 K=2,KN
      DO 10 I=2,IN1
      DO 10 J=2,JN1
      IF(MAR(I,J).EQ.11) GO TO 44
      GO TO 10
44    DHX=(HTE(I+1,J)-HTE(I-1,J))/(2*DX)
      DHY=(HTE(I,J+1)-HTE(I,J-1))/(2*DY)
      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
      DHT=(HTD(I,J)-HTE(I,J))/(DT)
      W(I,J,K)=(WH(I,J,K)-(((K-1)*DZ-1)*DHT-D(I,J,K)*ETAX*XX(I)
     C-E(I,J,K)*ETAY*YY(J)+(K-1)*DZ*D(I,J,K)*DHX*XX(I)+(K-1)*DZ*
     CE(I,J,K)*DHY*YY(J)))/HT(I,J)
10    CONTINUE
      RETURN
      END
```

7.2.27. WVEL

This subroutine computes the vertical velocity (Ω) by using the equation

$$\Omega = - \frac{1}{H} \int_{o}^{\sigma} \{ X' \frac{\partial (HU)}{\partial X} + Y' \frac{\partial (HV)}{\partial Y} \} d\sigma + \frac{\sigma}{H} \int_{o}^{1} \{ X' \frac{\partial (HU)}{\partial X} + Y' \frac{\partial (HV)}{\partial Y} \} d\sigma + \frac{\sigma}{H} (W_b - U_b X' \frac{\partial h}{\partial X} - V_b Y' \frac{\partial h}{\partial Y})$$

the numerical scheme used is central in space and trapezoidal rule is for numerical integration.

```
C     THIS PROGRAM CALCULATES THE VERTICAL VELOCITY IN THE GAMMA COORDS
C
      SUBROUTINE WVEL(IN,JN,KN,U,V,W,HT,DX,DY,DZ,MAR,XX,YY,WH)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),W(IN,JN,KN),MAR(IN,JN)
     C,XX(IN),YY(JN),WH(IN,JN,KN)
      KN1=KN-1
      IN1=IN-1
      JN1=JN-1
      DO 10 I=2,IN1
      DO 10 J=2,JN1
      DUM=0.
      DO 9 K=1,KN
      IF(MAR(I,J).EQ.11) GO TO 20
      GO TO 10
20    D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2*DY)
      IF (K.EQ.1) GO TO 17
      IF(K.EQ.KN) GO TO 17
      DUM=DUM+DZ*(D1HUX*XX(I)+D1HVY*YY(J))/HT(I,J)
      GO TO 9
17    DUM=DUM+DZ*(D1HUX*XX(I)+D1HVY*YY(J))/(2*HT(I,J))
9     CONTINUE
      DUM=DUM+W(I,J,KN)
      WUD=0.
      DO 8 K=2,KN1
      IF(MAR(I,J).EQ.11) GO TO 40
      GO TO 10
40    D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2*DX)
      D1HUY=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2*DX)
      D1HVY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2*DY)
      D1HVY1=(HT(I,J+1)*V(I,J+1,K-1)-HT(I,J-1)*V(I,J-1,K-1))/(2*DY)
      WUD=WUD+DZ*(D1HUX*XX(I)+D1HUX1*XX(I)+D1HVY*YY(J)+D1HVY1*YY(J))/
     C(2*HT(I,J))
      W(I,J,K)=-WUD+DUM*(K-1)*DZ
8     CONTINUE
10    CONTINUF
      RETURN
      END
```

7.2.28. XYSH

This subroutine does the stretching in both horizontal directions and determines the constants XX(I), YY(J), XXX(I) and YYY(J). This subroutine needs the values of DEEX, DEEY, EEEX which are read in the main program. In order to obtain these values, another main program "CONST" has to be run.

```
C     THIS SUBROUTINE COMPUTES THE HORIZONTAL STRETCHING CONSTANT
C
      SUBROUTINE XYSH(IN,JN,DELX,DELY,DEEX,DEEY,EFEX,EEEY,XX,YY,
     CXXX,YYY,A,B,X,Y)
      DIMENSION X(IN),XX(IN),XXX(IN),Y(JN),YY(JN),YYY(JN),A(IN),B(JN)
      DO 10 I=1,IN
      X(I)=(I-1)*DELX
      A(I)=(X(I)-DEEX)/EEFX
      XX(I)=1./COSH(A(I))
      XXX(I)=-SINH(A(I))/(EEEX*COSH(A(I))**3)
   10 CONTINUE
      DO 20 J=1,JN
      Y(J)=(J-1)*DELY
      B(J)=(Y(J)-DEEY)/EEEY
      YY(J)=1./COSH(B(J))
      YYY(J)=-SINH(B(J))/(EEEY*COSH(B(J))**3)
   20 CONTINUE
      RETURN
      END
```

REFERENCES

Crandell, S. H., Engineering Analysis, McGraw Hill, New York, 1965.

Edinger, J. E., and Geyer, J.C., "Heat Exchange in the Environment," E.E.I. Publication No. 65-902, Edison Electric Institute, 1965. (R1971)

Harleman, D.R.F., and Stolzenbach, K.D., "Engineering and Environmental Aspects of Heat Disposal from Power Generation," Dept. of Civil Engineering, M.I.T., June 1975.

Lee, S. S., and Sengupta, S., "Three-Dimensional Thermal Pollution Models," Phase III, NASA-CR-144858, 1976.

Murrill, P.W., and Smith, C.L., An Introduction to FORTRAN IV Programming, Intext Educational Publishers,1975.

Sengupta, S., Lee, S.S., and Miller, H.P., "A Three-Dimensional Free-Surface Numerical Model for Transport Processes in Biscayne Bay", in Computing Methods in Geophysical Mechanics, ASME Applied Mechanics Division, 1977.

Tsai, C.F., Ph.D. dissertation, University of Miami, Coral Gables, Florida, 1977.

Waldrop, W., and Farmer, R.C., "Three-Dimensional Flow and Sediment Transport at River Months," Coastal Studies Institute, Louisiana State University, Baton Rouge. Technical Report No. 150, Sept. 1973.

C-3

TABLE 1: Representation of variables at different time levels

VARIABLE	n-1	n	n+1
Height (H)	HT	HTD	HTE
Height (h)	HI	HI	HI
u-velocity	U	H	D
v-velocity	V	G	E
w-velocity	WH	WH	WH
Ω	W	W	W
Density	RR	RR	RR
Temperature	T	TN	TF
η	ETA	ETA	ETA

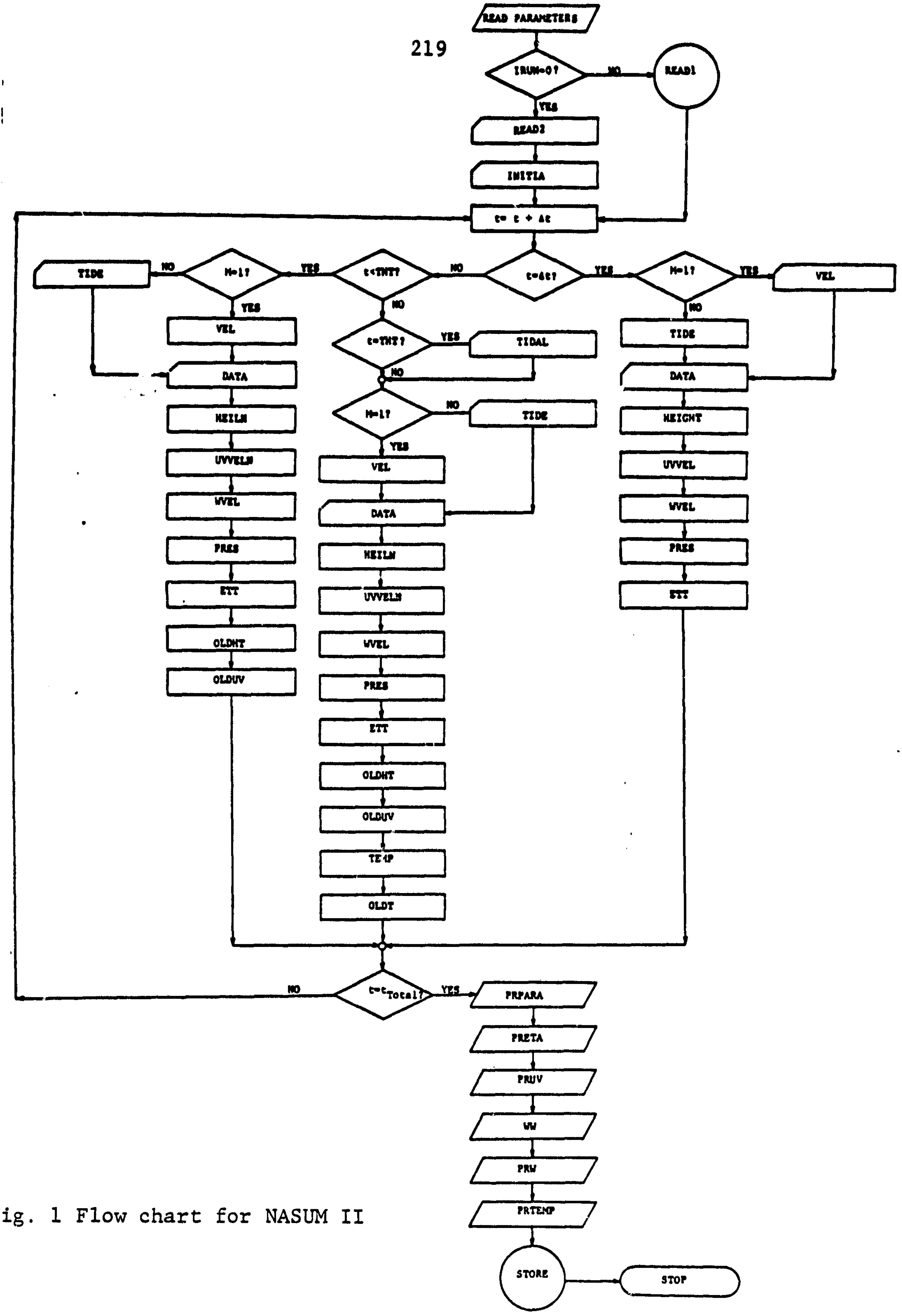


Fig. 1 Flow chart for NASUM II

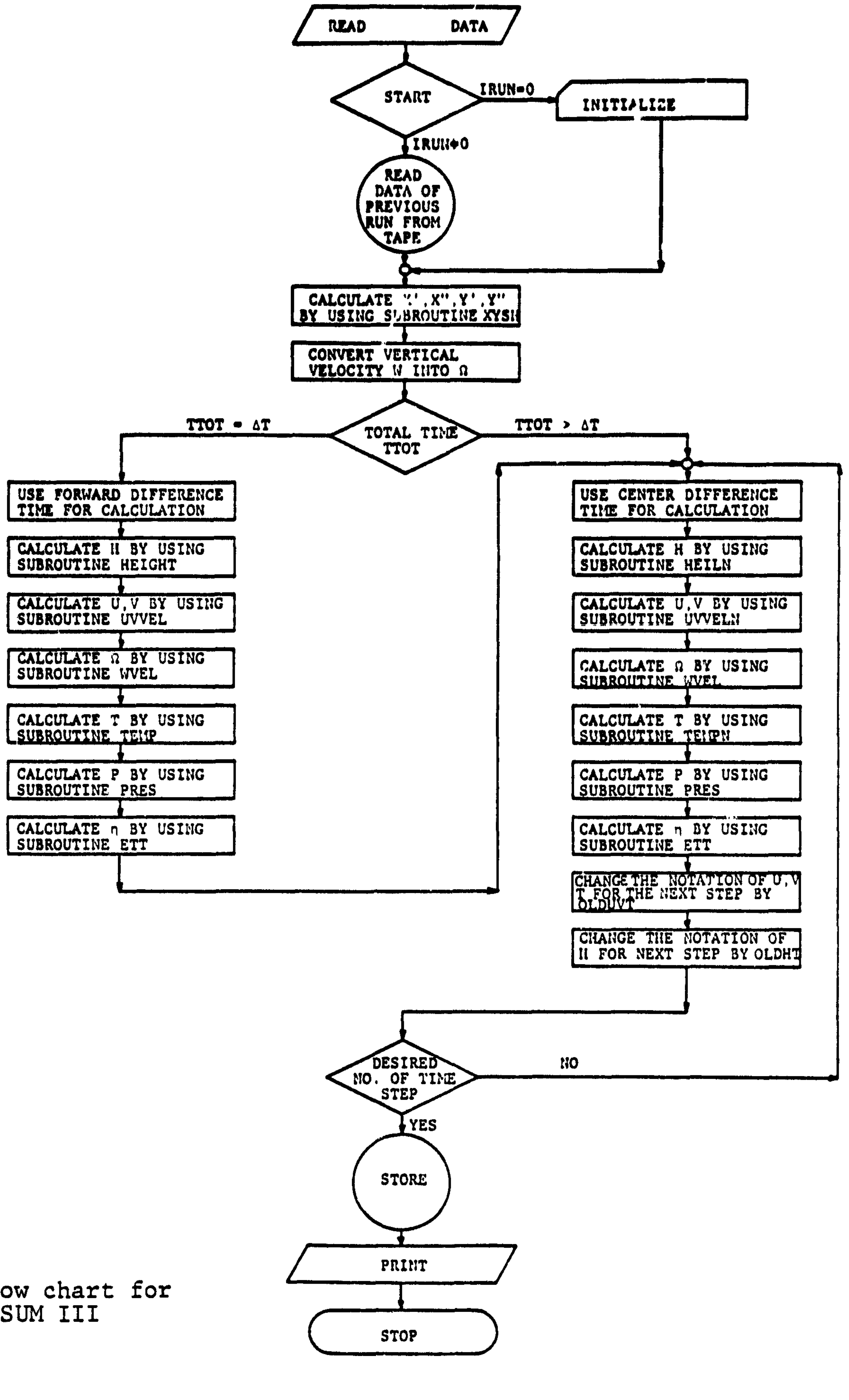

READ DATA
START
IRUN=0
INITIALIZE
IRUN≠0
READ DATA OF PREVIOUS RUN FROM TAPE
CALCULATE X',X",Y',Y" BY USING SUBROUTINE XYSH
CONVERT VERTICAL VELOCITY W INTO Ω
TTOT = ΔT
TOTAL TIME TTOT
TTOT > ΔT
USE FORWARD DIFFERENCE TIME FOR CALCULATION
CALCULATE H BY USING SUBROUTINE HEIGHT
CALCULATE U,V BY USING SUBROUTINE UVVEL
CALCULATE Ω BY USING SUBROUTINE WVEL
CALCULATE T BY USING SUBROUTINE TEMP
CALCULATE P BY USING SUBROUTINE PRES
CALCULATE η BY USING SUBROUTINE ETT
USE CENTER DIFFERENCE TIME FOR CALCULATION
CALCULATE H BY USING SUBROUTINE HEILN
CALCULATE U,V BY USING SUBROUTINE UVVELN
CALCULATE Ω BY USING SUBROUTINE WVEL
CALCULATE T BY USING SUBROUTINE TEMPN
CALCULATE P BY USING SUBROUTINE PRES
CALCULATE η BY USING SUBROUTINE ETT
CHANGE THE NOTATION OF U,V T FOR THE NEXT STEP BY OLDUVT
CHANGE THE NOTATION OF H FOR NEXT STEP BY OLDHT
DESIRED NO. OF TIME STEP
NO
YES
STORE
PRINT
STOP

Fig 2 Flow chart for NASUM III

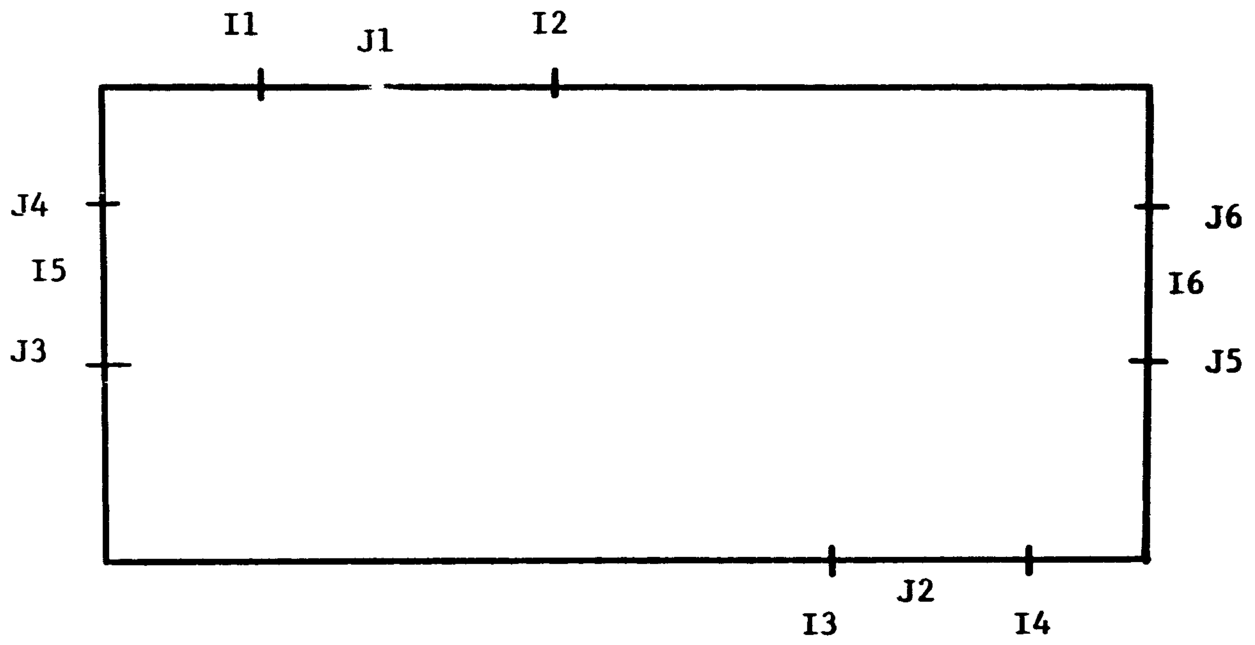


Fig.3 Location of Open Boundaries for NASUM II (Far-Field)

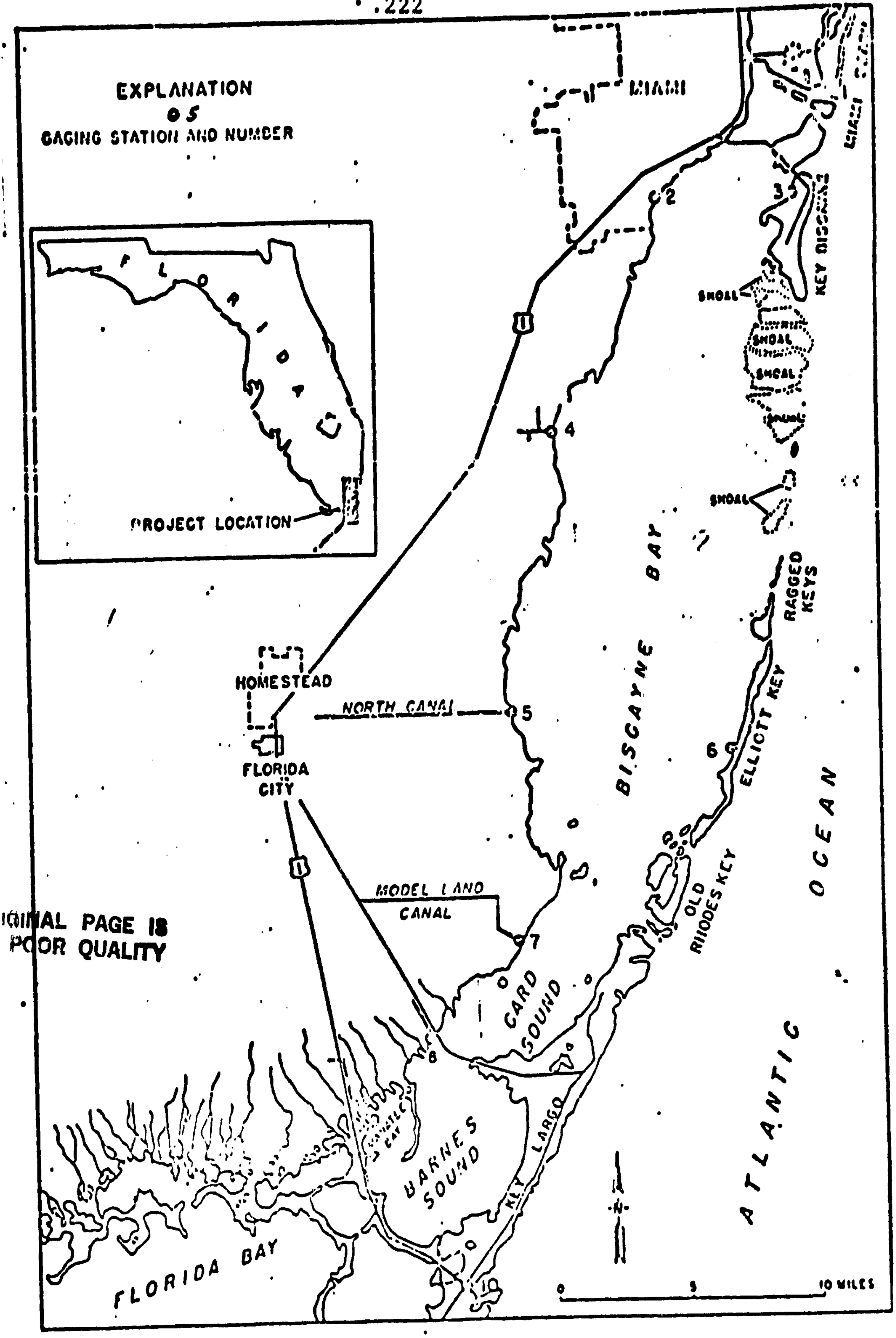


Fig. 4 Map of southeastern Dade County showing the area of investigation and location of gaging stations.

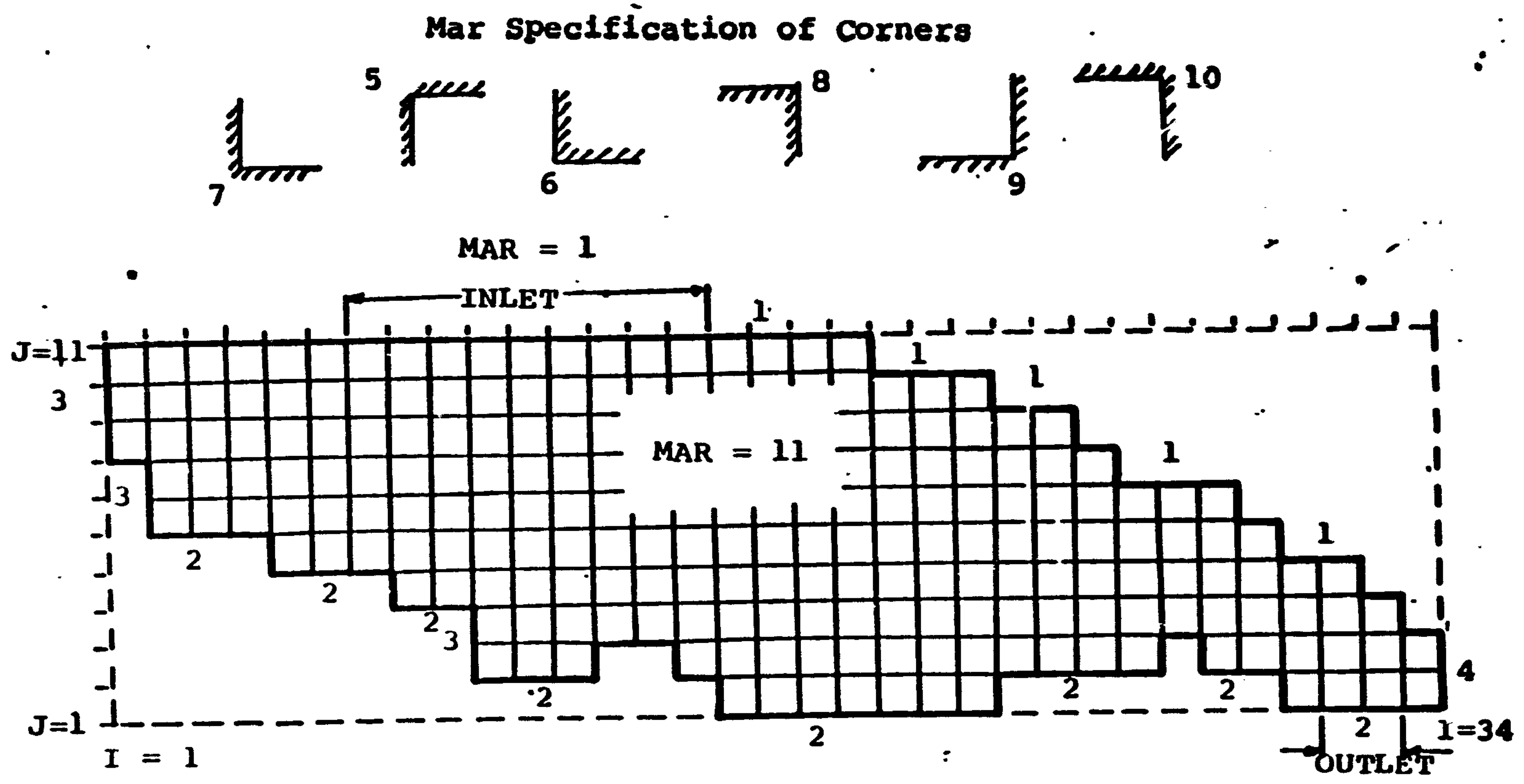


Fig. 5 Grid configuration and MAR node values.

THERMAL POLLUTION LAB
UNIVERSITY OF MIAMI

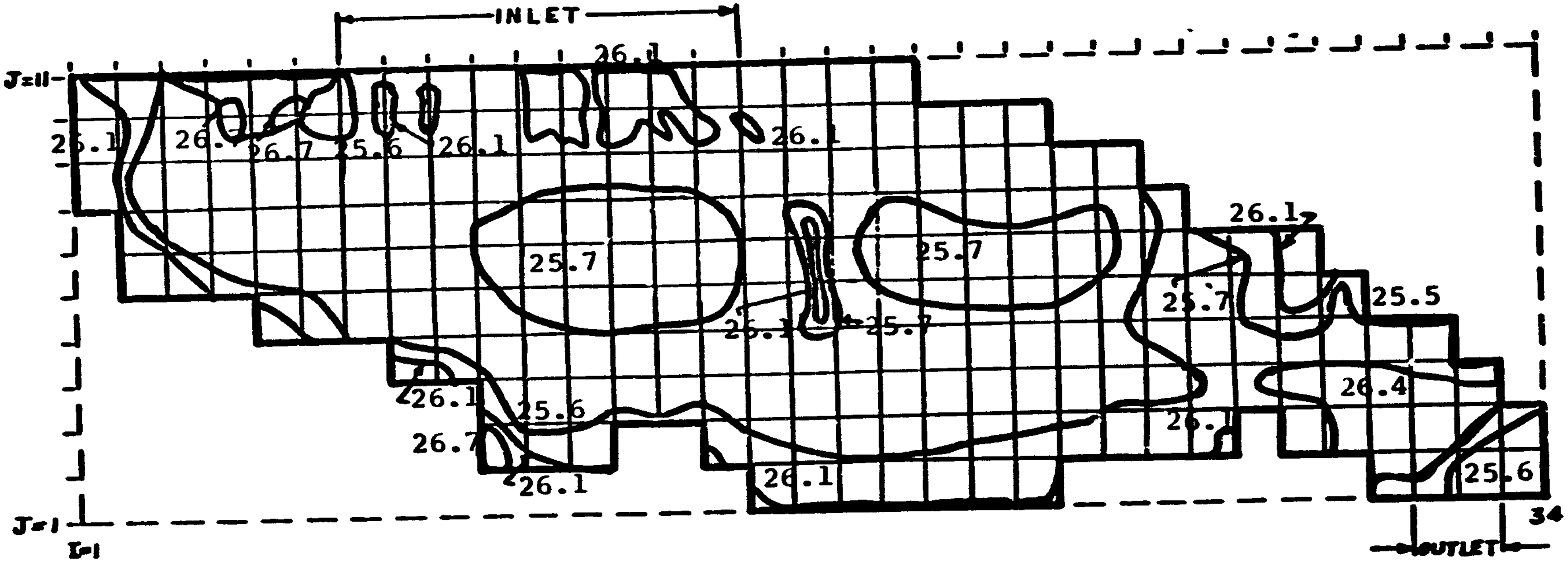


Fig.6 Surface Isotherms For Biscayne Bay From NASA - 6 IR Scanner Data, Corrected by Ground Truth, on the morning of April 15, 1975

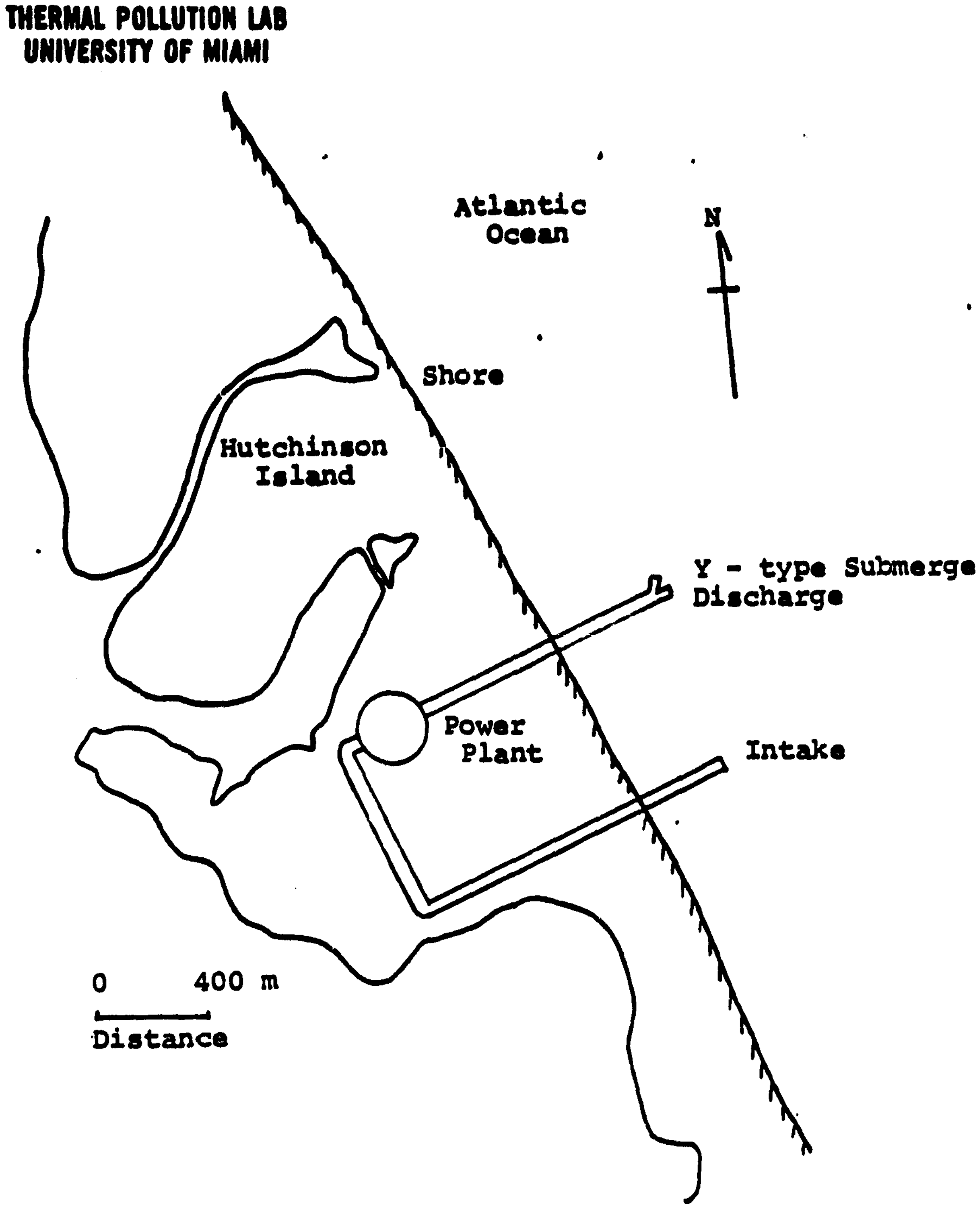


Fig. 7 Florida Power and Light Companys Hutchinson Island Site Power Plant

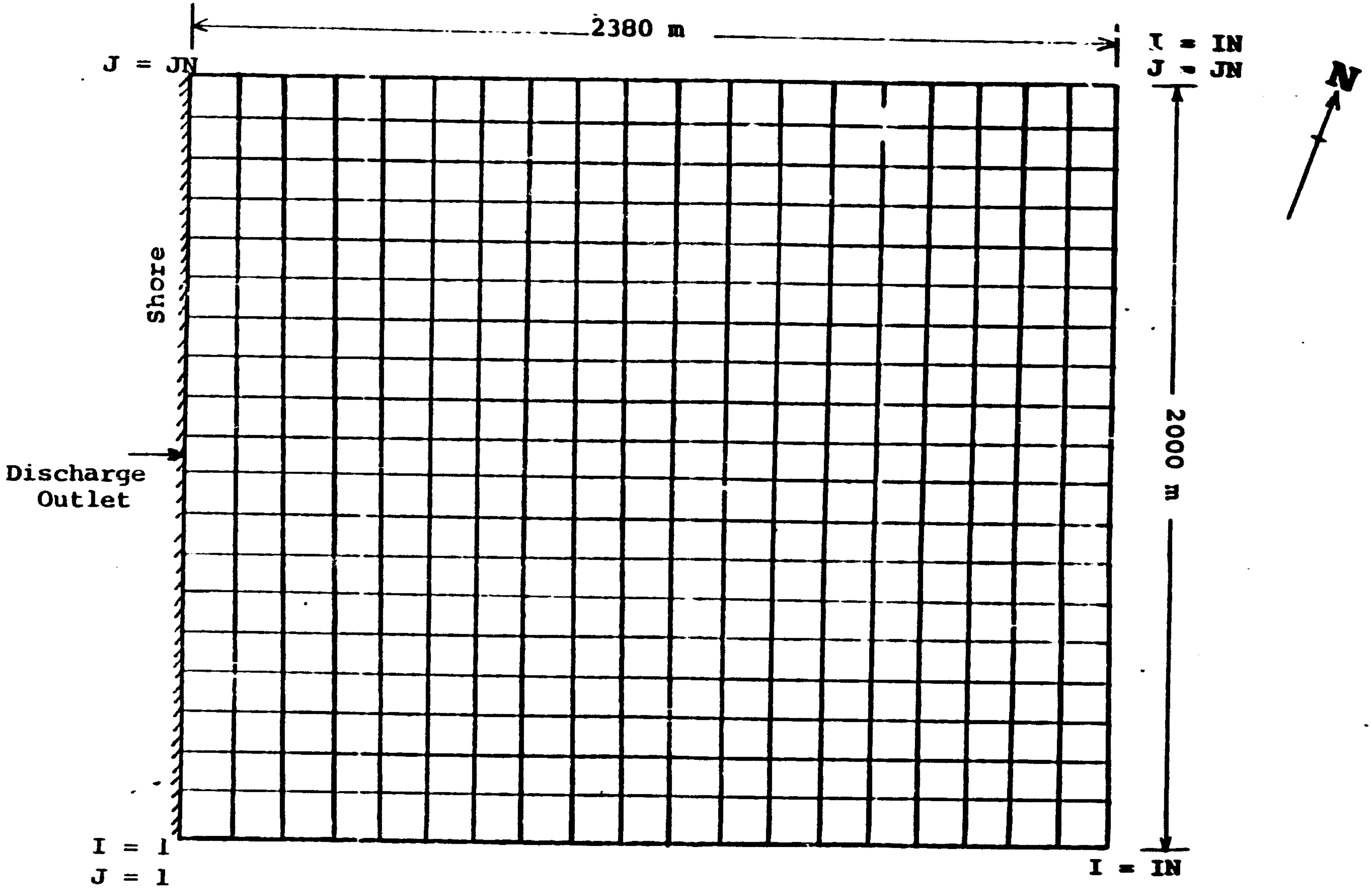


Fig. 8 Horizontal Grid Point System Without Stretching For Free Surface Near Field Model Applied to Hutchinson Island Site

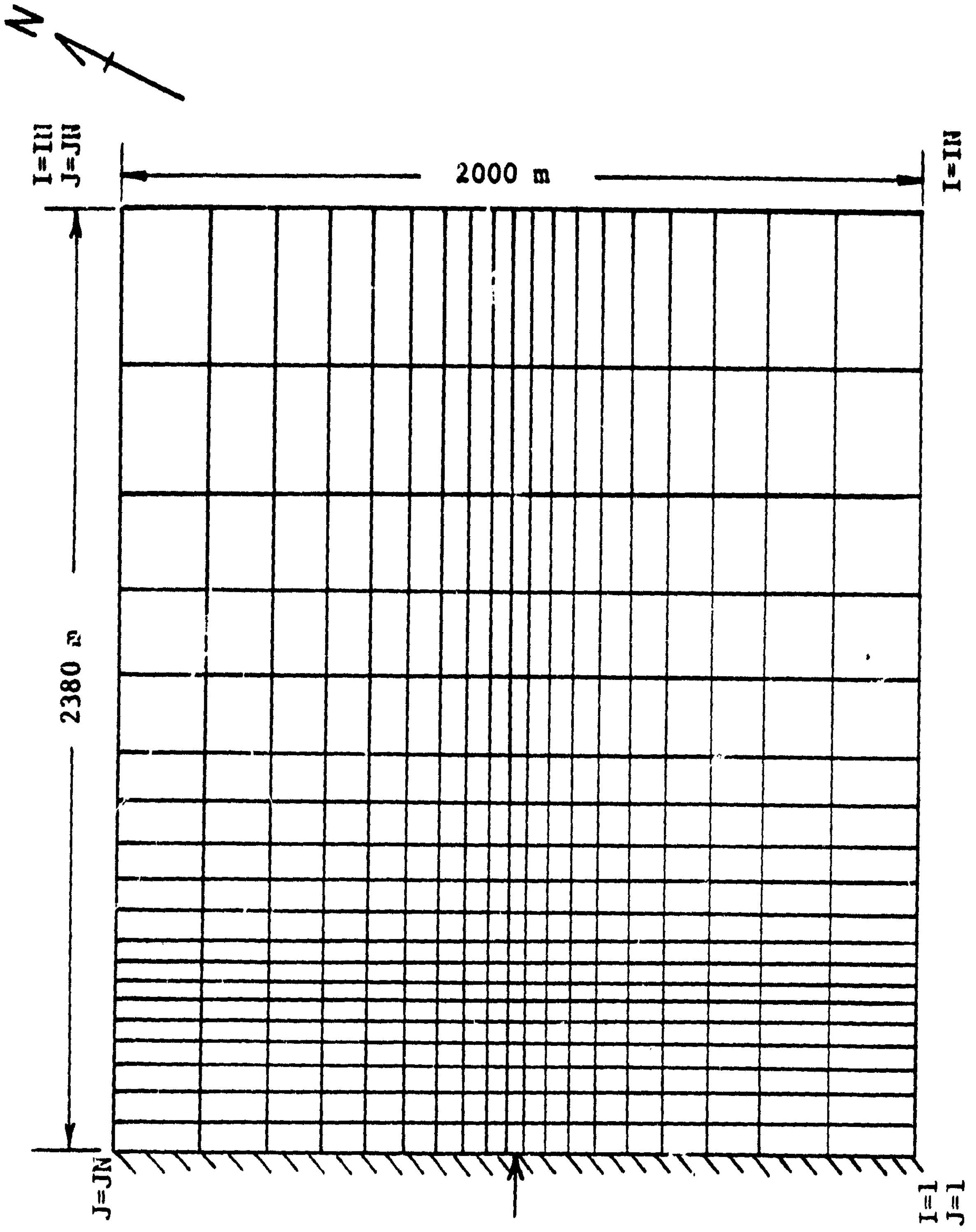


Fig.9 Physical horizontal grid point system for the free surface model sample problem applied to Hutchinson Island site.

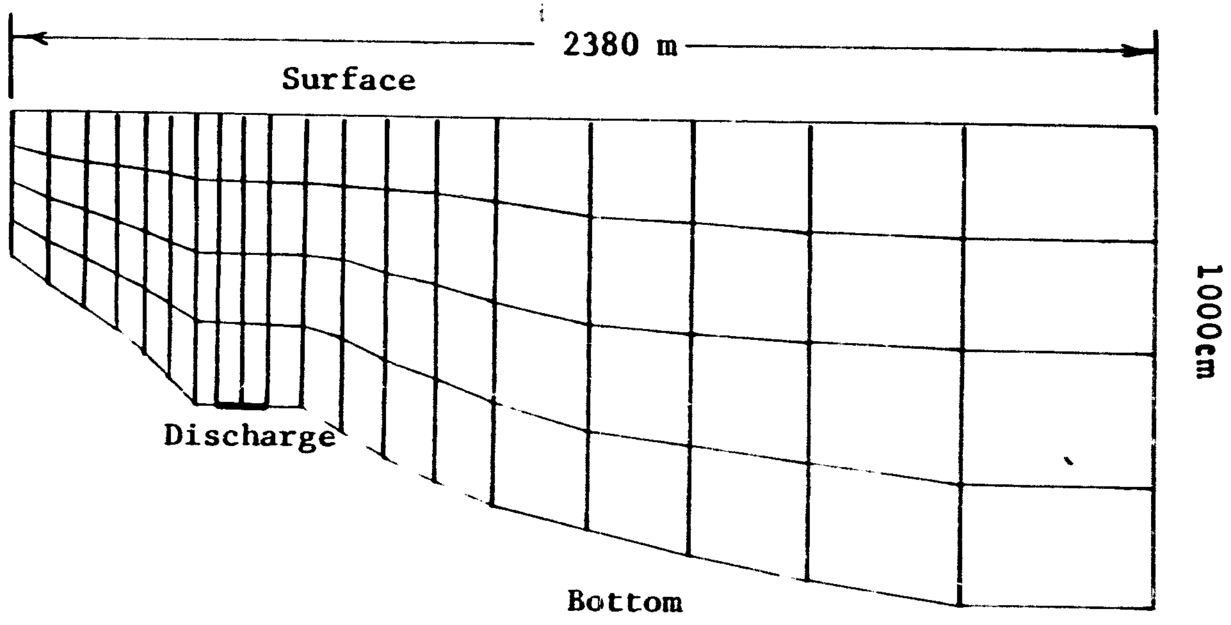
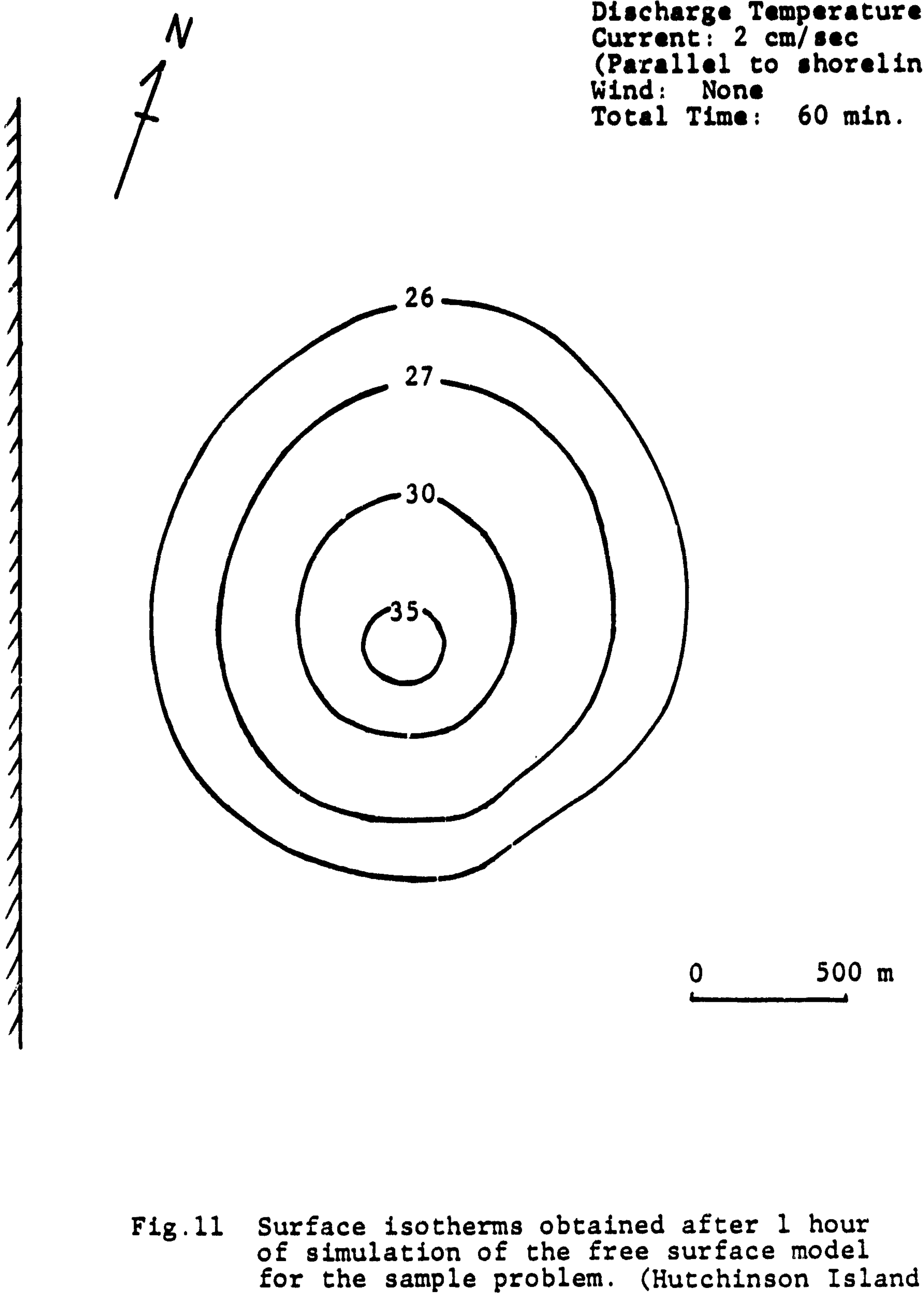


Fig.10 Numerical grid point system on a distorted vertical section for the free surface model sample problem applied at Hutchinson Island.



Discharge volume 263,000 G.P.M
Discharge Velocity: 0.102 cm/sec
Discharge Temperature: 35.0°C
Current: 2 cm/sec
(Parallel to shoreline)
Wind: None
Total Time: 60 min.

Fig.11 Surface isotherms obtained after 1 hour of simulation of the free surface model for the sample problem. (Hutchinson Island Site)

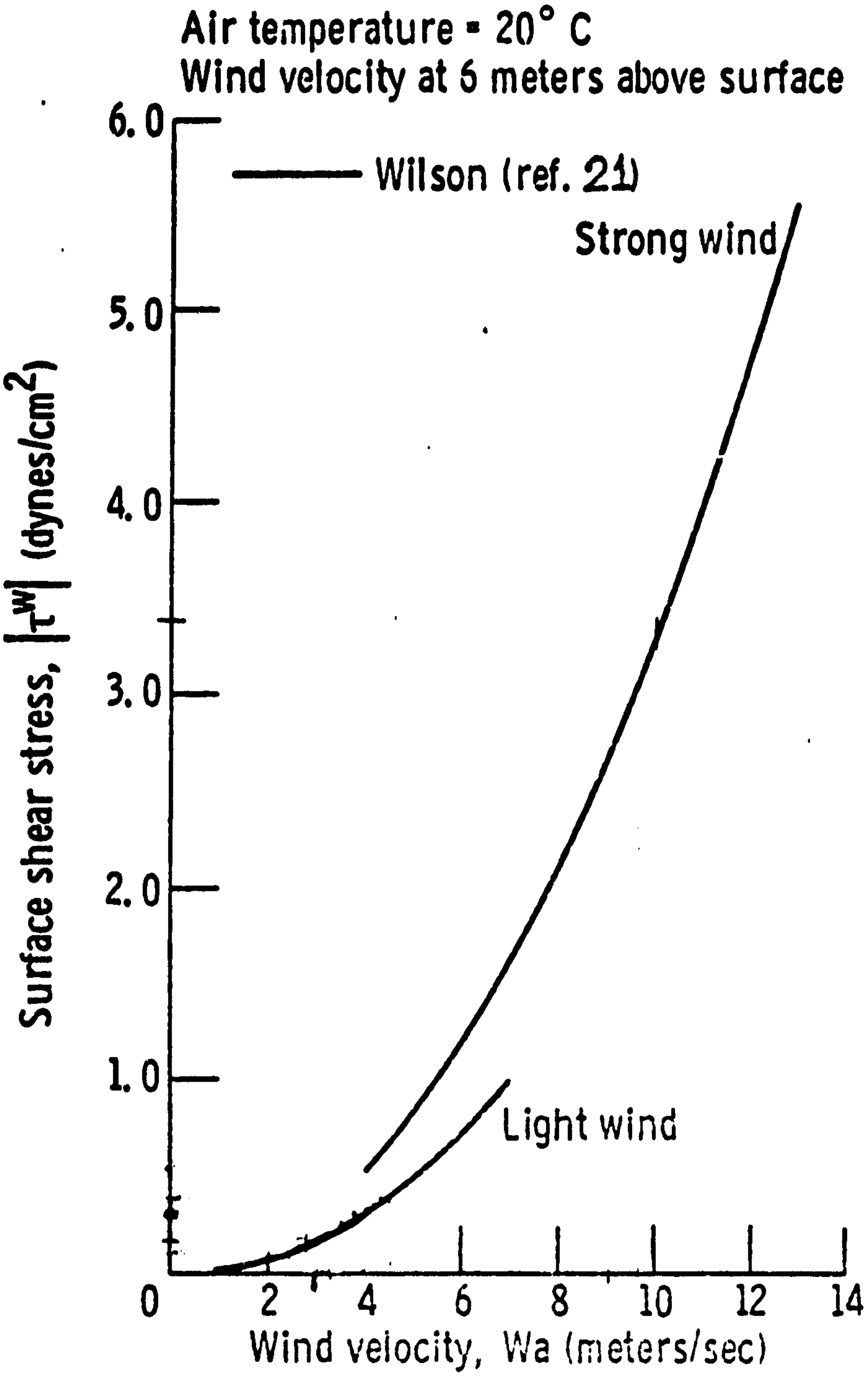


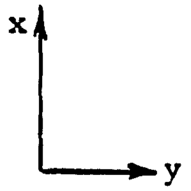
Figure 12 Wind shear stress relation.

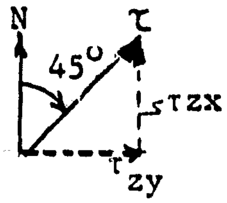
APPENDIX A

Wind Stresses

The wind shear stresses τ_{zx} and τ_{zy} are computed by using the Wilson curve as shown in Fig.12. First, the magnitude of the wind velocity, in meters/sec., is used to read off from this curve the resultant shear stress; τ_{zx} and τ_{yz} are determined by simply resolving τ into its respective components.

As an example, consider a wind of 10 mph from the South West direction. Assume that the direction of North is in the same direction as the positive x-axis, and East is in the same direction as the positive y-axis as shown below.





Then $\tau_{zx} = \tau \cos 45^\circ = .4 \cos 45^\circ = .283$ dynes/cm^2

$\tau_{zy} = \tau \sin 45^\circ = .4 \sin 45^\circ = .283$ dynes/cm^2

Where $\tau = .4$ dynes/cm^2 for 10 mph = 4.47 meters/sec.

APPENDIX B

HEAT TRANSFER MECHANISMS

The analysis in this section is taken from Harleman et al. (1973)

1. Solar Radiation (short wave)

The incident solar radiation impinging on the water surface may be expressed as:

$$\varphi_s = \varphi_{sc}(1-0.65C^2)$$

Where ψ_{sc}= clear sky solar radiation obtained using the 100% possible sunshine curve (given in Appendix B)

C = fraction of sky covered by clouds

The reflected solar radiation is typically 6% of incident solar radiation, hence the net solar radiation absorbed by the water surface is:

$$\varphi_{sn} = \varphi_s - \varphi_{sr} \simeq 0.94\varphi_{sc}(1-0.65C^2)$$

2. Atmospheric Radiation (long wave)

The basic equation for the incident atmospheric radiation, φ_a is given as:

$$\varphi_a = \epsilon\sigma T_a^{*4}$$

Where ϵ = average emmitance of the atmosphere

σ = Stefan-Boltzmann constant

$T_a{}^*$ = air temperature (absolute)

However, good agreement with experimental data has indicated that is a function of T_a ( ), and specifically, T_a*6 dependence gives best results for atmospheric radiation at low temperatures, as well as providing a good fit at high temperatures. Clear sky incident atmospheric radiation, φ_{ac}, may be expressed as:

$$\varphi_{ac} = 1.2 \times 10^{-13}(T_a^*)^6$$

and, then incident atmospheric radiation including cloudiness may be expressed as:

$$\varphi_a = \varphi_{ac}(1 + 0.17C^2)$$

A figure of 3% is usually accepted as reflectance of a water surface to longwave radiation. Thus the net atmospheric radiation absorbed by the surface is:

$$\varphi_{an} = \varphi_a - \sigma_{ar} = 0.97\varphi_a$$

and, therefore, we have:

$$\varphi_{an} = 1.16 \times 10^{-13}(T_a^*)^6(1 + 0.17C^2)$$

3. Longwave Radiation from the Water Surface φ_{br}

In reference ( ) it is noted that the emmissivity of a water surface is independent of temperature and salt or colloidal concentrations, and gives a value of 0.97. Thus we obtain:

$$\varphi_{br} = 0.97\sigma(T_s^*)^4$$

Where T_s = water surface temperature.

4. Evaporative Heat Flux, φ_e

Evaporation from a water surface occurs as a result of both forced (wind driven) convection and free (bouyancy driven) convection. The evaporation from a water surface is usually written (mass/area/time) as:

$$E = \rho F(W_z)(e_s - e_a)$$

Where, E = mass flux (mass/area/time)

ρ = density of water

W_z = windspeed at height z above surface

$F(Wz)$ = windspeed function for mass flux including both free and forced convection effects (length/time/pressure)

e_s = saturated vapor pressure at T_s

e_z = vapor pressure at height z above surface

Then writing the above equation in heat units, the evaporative heat flux, φ_e is given by:

$$\varphi_e = F(W_z)(e_s - e_z)$$

Where $F(W_z)$ = windspeed function for heat flux (energy/area/time/pressure)

Now, dropping the z subscript (and assuming W measured "z" above the surface $\approx$ W at the surface) we may expres F(W) for a natural water surface and for an artificially heated water surface as:

$F(W) = 17W$ . . . natural water surface

and $F(W) = 22.4(T_s - T_a)^{1/3} + 14W$ . . . artificially heated surface

surface

5. Conduction Heat Flux,

Bowen (see reference) has suggested that conduction can be directly related to evaporative fluxes by assuming that eddy diffusivities of heat and mass are identical. Thus,

$$\varphi_c = R_b \varphi_e$$

where $R_b = C_b \cdot \left|\frac{T_s - T_a}{e_s - e_a}\right|$ = Bowen Ratio

and C_b = Bowen constant = 0.255 mm Hg/$^\circ$F

and, therefore the conduction heat flux, φ_c, may be expressed as,

$$\varphi_c = C_b F(W) (T_s - T_a)$$

APPENDIX C

THE EQUILIBRIUM TEMPERATURE AND THE SURFACE HEAT TRANSFER COEFFICIENT

The net heat transfer ( ) through a water surface is composed of radiation penetrating the water surface from above, radiation out of the water surface, evaporation, and conduction transfer. These are indicated schematically in the following figure:

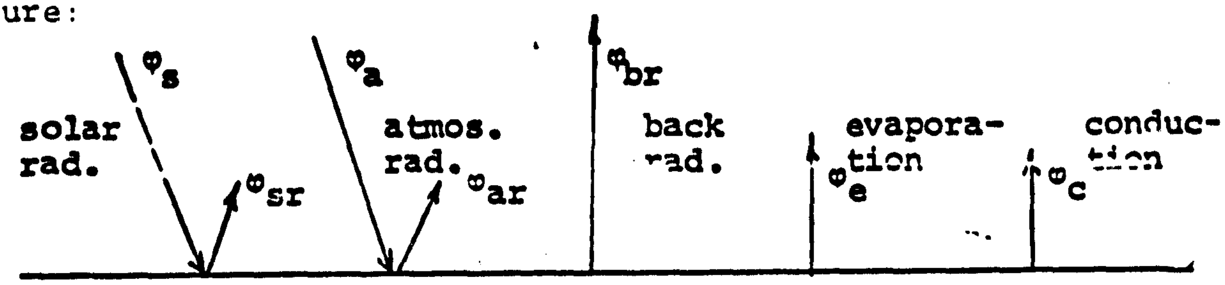


Fig. F-1. Heat Transfer Mechanisms at the Water Surface.

The following heat balance results,

$$\varphi_n = \underbrace{\varphi_s - \varphi_{sr}}_{\varphi_{sn}} + \underbrace{\varphi_a - \varphi_{ar}}_{\varphi_{an}} - \varphi_{br} - \varphi_e - \varphi_c \text{ ....(F-1a)}$$

where φ_n = net heat input = $\varphi_{sn} + \varphi_{an} - \varphi_{br} - \varphi_e - \varphi_c$ ...(F-1b)

Now, equation (F-1) may be rewritten as,

$$\varphi_n = \varphi_r - \varphi_L \text{ ....(F-2)}$$

Where φ_r = net absorbed radiation = $\varphi_{sn} + \varphi_{an}$

and $\varphi_L = \varphi_{br} + \varphi_e + \varphi_c$

A. Equilibrium Temperature Calculation, Te (See Appendix E)

Under equilibrium conditions equation (F-2) yields,

$\varphi_n = 0 = \varphi_r - \varphi_L$

so that,

$$\varphi_r = \varphi_L \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(F-3)}$$

Then by using the approximate formulae in reference ( ) we obtain by setting $T_s = T_e$,

$$0.94\ \varphi_{sc}(1-0.65C^2) + 1.16 \times 10^{-13}(T_a*)^6(1+0.17c^2)$$
$$= 0.97\varphi(T_e*)^4 + F(W)\left[(e_s-e_a) + C_b(T_e-T_a)\right] \ldots \quad \text{(F-4)}$$

Where φ_{sc} = clear sky solar radiation

C = cloudiness ratio

T_a = air temperature (°C or °F)

T_e = equilibrium temperature (°C or °F)

$T*$ = absolute temperature (°K or °R)

$F(W)$ = windspeed function (BTU/ft^2/day, mm Hg)

e_s = saturated vapor pressure at water surface temperature (mm Hg)

e_a = saturated vapor pressure at air temperature (mm Hg)

σ = Stefan-Boltzmann constant $\cong 4.1 \times .0^{-8}$ BTU/ft^2, day, °R^4

C_b = Bowen constant = 0.255 mm Hg/°F (see Appendix E)

W = windspeed (mph)

For a natural water surface,

$$F(W) = 17W \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(F-5a)}$$

and, for an artificially heated surface,

$$F(W) = 22.4\ (T_e-T_a)\ 1/3 + 14W \ldots\ldots\ldots\ldots \quad \text{(F-5b)}$$

Thus, equation (F-4) becomes,

$$0.94\ \varphi_{sc}(1-0.65C^2) + 1.16 \times 10^{-13}(T_a{*})^6(1+0.17C^2)$$

$$= 0.97\varphi(T_e{*})^4 + 17W\ [(e_s-e_a) + 0.255(T_e-T_a)]$$

or,

$$0.94\ \varphi_{sc}(1-0.65C^2) + 1.16 \times 10^{-13}(T_a{*})^6(1+0.17C^2)$$

$$= 0.97\varphi(T_e{*})^4 + [\ 22.4(T_e-T_a)^{1/3} + 14W]$$

$$[(e_s-e_a) + 0.255(T_e-T_a)\ ]$$

Therefore for known φ_{sc}, e_s, c_a, T_a and $W \rightarrow T_e$ can be determined by trial and error methods.

B. Surface Heat Transfer Coefficient (K)

From reference ( ) the surface heat transfer coefficient K, can be determined as follows,

$$K = \frac{\partial\varphi_L}{\partial T_{av}} = \frac{\partial\varphi_n}{\partial T_{av}}\text{, since } \varphi_r \neq \varphi_r(T_s) \text{ and } \therefore \frac{\partial\varphi_r}{\partial T_{av}} = 0.$$

where $T_{av} = (T_s + T_e)/2$

Thus,

$$K = 3.88\varphi(T_s^*)^3 + F(W)[(\frac{\partial e_s}{\partial T})_{T=T_{av}} + C_b]$$

$$+ [(e_s - e_a) + C_b(T_s - T_a)]\ \frac{\partial F(W)}{\partial T_{av}}$$

Where $\frac{\partial F(W)}{\partial T_{av}} = \begin{cases} 0 & \text{for natural water surface} \\ 1/3(22.4)(T_s-T_a)^{-2/3} & \text{for artificially heated water surface} \end{cases}$

C. Numerical Example

Consider natural water surface

$C = 0$

$T_a = 25^\circ C$

$W = 10$ mph

Location - Miami (latitude $26^\circ N$)

Date - December 20

From reference ( ),

$$@T_a = 25^\circ C \rightarrow e_a \cong 0.43 \text{ psia}$$

$$@T_e = 27^\circ C \; \underline{\text{as quess}} \cong e_s = e_e \cong 0.51 \text{ psia}$$

From reference ( ), Figure 2.15, pg. 2-61 (see Figure F-3)

$$\varphi_{sc} \cong 425 \text{ Langleys/day} = 1560 \text{ BTU/ft}^2\text{/day} \ldots \text{ using}$$

100%sunshine curve at $26^\circ N$, Dec. 20

Note: 1 Langley/min. = 220.62 BTU/ft^2, hr. = 1 calarie/cm^{2}min.

Then using equation (F-6) with $C = 0$,

$$0.94\ (1560)\ (1) + 1.16 \times 10^{-13}(5.37\times10^2)^6(1) = 4250$$

$$4 \times 10^{-8}(5.406\times10^2)^4 + 170\,[\,(.255)\,(2)+(.08)\,(51.7)]$$

4206 close enough!

$$\therefore \; T_e \cong 27^\circ C$$

(where 1 psia = 51.7 mm Hg)

Then from equation ( ),

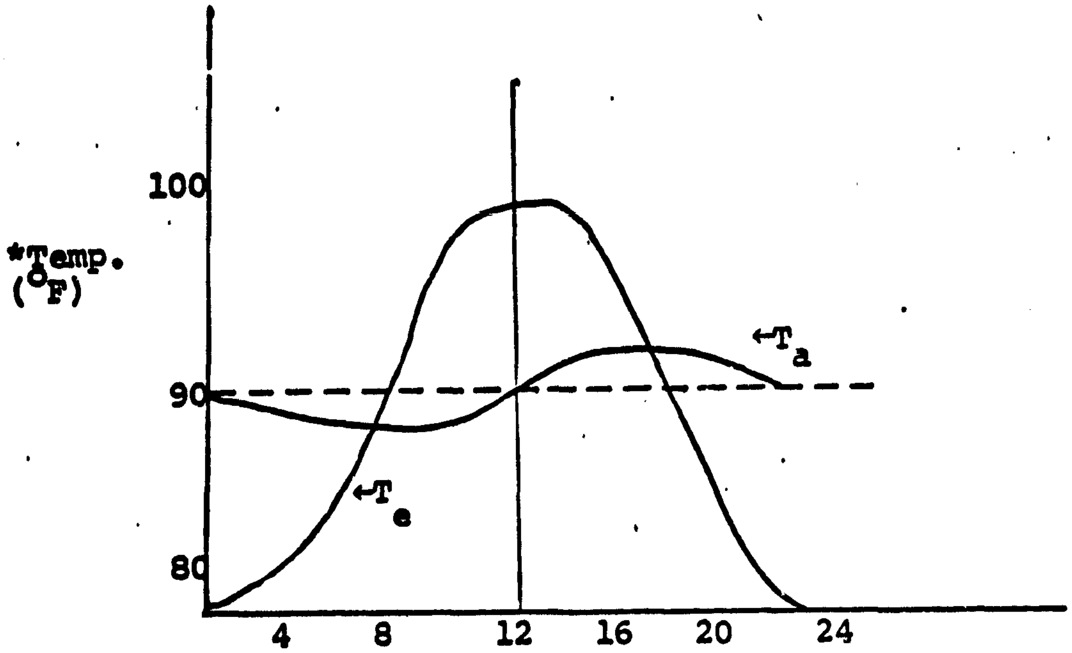
$$K = 3.88 \times 4.1 \times 10^{-8}(5.406\times10^2)^3 + 170\,(.255+0.0251\times51.7)$$

$$K \cong 290 \text{ BTU/ft}^2, {}^\circ F, \text{ day}$$

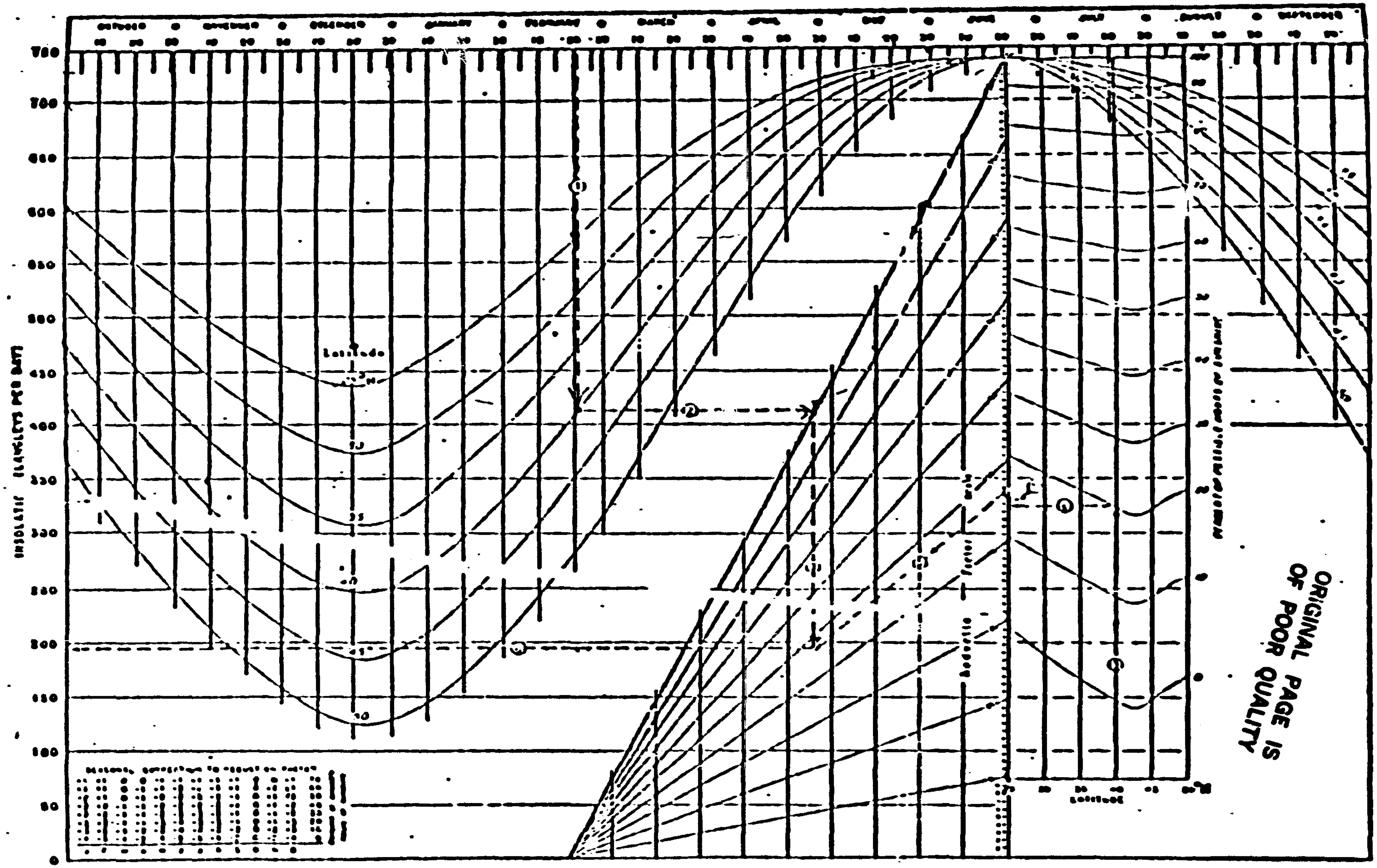
where $\left.\frac{\partial e_s}{\partial T}\right]_{T=T_{av}} \approx \frac{e_e - e_a}{T_e - T_a} = .0251$

D. Discussion

The equilibrium surface temperature, T_e , for a natural water surface, can be greater than the atmospheric temperature, T_a, whereby T_s increases from values below T_a up to T_e as equilibrium is reached. As can be seen in the figure below (F-2) T_e can be greater or smaller than T_a depending on the time of the day. Simply $T_e > T_a$ during the hours of sunshine and $T_e > T_a$ at night when the water surface is cooling.



*This plot is taken from (   ) and has no relation to the numerical example given in this paper. <u>However</u>, the numerical example considered 100% possible hours of sunshine ($T_e > T_a$).



Working diagram for computing insolation as a function of percent possible sunshine, latitude, and time of year. The example shown by dashed lines with arrows estimates the daily total insolation at a station at latitude 40° N., on February 21 and with 30 percent possible sunshine. Step 1.—enter the graph at February 21 and move downward, to the curve labeled 40° N. Step 2—then move horizontally to establish a reference point at the intersection with the heavy line. Step 3—enter lower right of the graph at latitude 40° N., proceed upward to 30 percent of possible hours of sunshine. Step 4—thence move horizontally toward the left to the vertical line through latitude 35° N. Step 5—here the seasonal correction (+2) from the inset table is added on the reduction factor scale. Step 6—thence move downward toward the convergence point. Step 7—from the reference point of step 2, move downward to an intersection with the line extended in step 6. Step 8—from the point of intersection of these two lines, move horizontally to the Q axis to read the estimated value, 300 langleys per day.

Figure F-3. Daily Average Isolation (from Hamon (1954))